The University of Mannheim
Institute of Computer Engineering

MOTION CONTROL OF HOLONOMIC WHEELED MOBILE ROBOT WITH MODULAR ACTUATION

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universitaet Mannheim

vorgelegt von
Ahmed Khamies El-Shenawy
aus Aegypten, Alexandria

Mannheim

15 April 2010

The thesis of Ahmed Khamies El-Shenawy was reviewed and approved* by the following:

Dekan:
Professor Dr. Felix Freiling,
Mannheim University

Referent:
Professor Dr. Essam Badreddin,
Heidelberg University

Korreferent:
Professor Dr. Reinhard Maenner
Heidelberg University

Tag der muendlichen Pruefung: 15. April 2010.

Abstract

This thesis proposes a control scheme for a new holonomic wheeled mobile robot. The platform, which is called C3P (**C**aster **3** wheels **P**latform), is designed and built by the Automation Lab., University of Heidelberg. The platform has three driven caster wheels, which are used because of their simple construction and easy maintenance.

The C3P has modular actuators and sensors configurations. The robot's actuation scheme produces singularity difficulties for some wheel steering configuration, described as the following: When all wheels yield the same steering angle value, the C3P cannot be actuated in the direction perpendicular to the wheel velocity vector. The C3P has a modular sensing scheme defined by sensing the steering angle and the wheel angular velocity of each caster wheel. This work has four main contributions

1- developing a controller based on an inverse kinematics solution to handle motion commands in the singular configurations;

2- modeling the C3P's forward dynamics of the C3P for the simulation purpose;

3- developing a motion controller based on an inverse dynamics solution; and

4- comparing the C3P with other standard holonomic WMRs.

In order to escape singularity condition, the actuated inverse kinematics solution is developed based on the idea of coupling any two wheel velocities to virtually actuate the steering angular velocity of the third wheel. The solution is termed as the **W**heel **C**oupling **E**quation (WCE). The C3P velocity controller consists of two parts: a) the WCE regulator to avoid singularities and adjust the steering

angles to the desired value, and b) the regular PID controller to maintain the reference robot velocities with respect to the floor frame of coordinates. The solution reaches acceptable performance in the simulation examples and in the practical experiments. However, it generates relatively large displacement errors only during the steering angles adjustment period.

The Euler-Lagrangian method is used for obtaining the forward dynamic and the inverse dynamic models. The forward dynamic model consists of two equations of motion: the WTD (**W**heel **T**orque **D**ynamics) to calculate the wheel angular velocities with respect to the actuated wheels' torques, and the DSE (**D**ynamic **S**teering **E**stimator) for calculating the steering angles and steering angular velocities corresponding to the angular wheels' velocities and accelerations.

The inverse dynamics solution defines the forces and torques acting on each actuator and joint. The solution is used in the development of the C3P velocity and position controllers. In comparison to the proposed inverse kinematics solution, the inverse dynamics solution yields less displacement errors. Lyapunov stability analysis is carried out to investigate the system stability for different steering angles' combinations. The steering angles' values are considered as the disturbances affecting the platform.

Finally, a comparison is made between the C3P and three other holonomic wheeled mobile robots configurations. The comparison is based on the simulation results in relation to the following aspects: a) mobility, b) total energy consumed by each robot in a finite interval of time and c) hardware complexity. The C3P platform shows its advantage in the aspects "b" and "c".

Table of Contents

List of Figures **ix**

List of Tables **xiii**

List of Symbols **xiv**

Chapter 1
Introduction **1**
1.1 Overview 1
1.2 Motivation 2
1.3 State of the Art 4
1.3.1 Kinematics Modeling 7
1.3.2 Dynamic Modeling 7
1.3.3 Wheeled Mobile Robot Control Structure 9
1.4 Problem Formulation 11
1.5 Main Contributions 13
1.6 Outline 14

Chapter 2
The C3P Kinematic and Dynamic Modeling **16**
2.1 Introduction 16
2.2 Kinematic Modeling 17
2.3 The C3P Kinematic Modeling 18
2.3.1 Inverse and Forward Kinematic Solutions 22
2.4 Robot Dynamics Modeling 25
2.4.1 Nonholonomically Constrained System 25
2.4.2 Holonomically Constrained System 26

2.4.3 The C3P Platform Constrained System . . . 27
2.4.4 Euler-Lagrange Method . . . 30
2.4.5 Kinetic Energy Equations . . . 31
2.5 Dynamic Modeling . . . 31
2.5.1 The Wheels Torque Dynamics (WTD) . . . 32
2.5.2 The Dynamic Steering Estimator (DSE) . . . 34

Chapter 3
Kinematics Based Motion Control **37**
3.1 C3P Singularities . . . 37
3.2 Coupling Approach . . . 39
3.2.1 Simulation Examples . . . 43
3.2.2 Singularity Indicator . . . 47
3.3 Wheel Coupling Equation Adaptation . . . 48
3.4 Velocity Controller . . . 50
3.5 Position Controller . . . 53
3.6 Summary . . . 56

Chapter 4
Inverse Dynamics Based Motion Control and Analysis **59**
4.1 Inverse Dynamics Solution . . . 59
4.2 Dynamics Based Motion Control Structure . . . 61
4.2.1 Velocity Controller . . . 62
4.2.2 Dynamics Performace Examples . . . 63
4.2.3 Position Controller . . . 71
4.3 Summary . . . 77

Chapter 5
The Lyapunov Stability Analysis **78**
5.1 Introduction . . . 78
5.2 The Lyapunov Function . . . 78
5.3 Numerical Analysis . . . 83
5.4 Conclusion . . . 86

Chapter 6
Implementation and Practical Results **89**
6.1 Platform Hardware Configuration . . . 89
6.2 Kinematics Based Controller Experiments . . . 91
6.3 Dynamic Based Control Results . . . 99
6.4 Experiments on C3P Stability . . . 111

6.5 Summary 114

Chapter 7
Comparing Different Holonomic WMRs **116**
7.1 Introduction 116
7.2 Description of Holonomic Mobile Robots 117
7.3 Comparing The C3P Vs Holonomic WMRs 119
7.3.1 Driving in 3 Degrees of Freedom 121
7.3.2 Driving in the Infinity Shape(∞) 122
7.4 Performance Function Comparison 126
7.4.1 Mobility Aspect 126
7.4.2 Energy Consumption Aspect 126
7.4.3 Hardware Complexity Aspect 127
7.4.4 Cost Functional Calculation 128
7.4.4.1 Driving in 3DOFs 129
7.4.4.2 Driving in (∞) Shape 131
7.5 Summary 132

Chapter 8
Conclusions and Future Work **134**

Appendix A
Kinematics Modeling **137**
A.1 The Velocity Generalized Wheel Jacobian 137
A.1.1 The Acceleration Wheel Jacobian 138
A.1.2 Actuated Inverse and Sensed Forward Kinematics 139

Appendix B
The Dynamic Steering Estimator (DSE) **141**

Appendix C
Inverse Dynamics Equations **143**
C.1 The Inverse Dynamics Solution 143
C.2 The Inverse Kinematics for Castor Wheel Acceleration Variables . . 144

Appendix D
The Lyapunov Analysis **149**

Appendix E
The Kinematics and Dynamics Modeling of Different Holonomic Wheeled Mobile Robot **154**

E.1 Kinematics Modeling of Holonomic Mobile Robots 154
E.1.1 Holonomic Caster Wheeled Robot (HCWR) 154
E.1.2 Omni Directional Wheeled Robot (ODWR) 155
E.1.3 Ramsis II 157
E.2 The Robots Dynamics Equations 158
E.2.1 Holonomic Caster Wheeled Robot 159
E.2.2 Omni Directional Wheeled robot 161
E.2.3 Ramsis II 162

List of Figures

1.1 Using WMR as explosives transporter [9] . . . 2
1.2 Rigid body degrees of freedoms . . . 2
1.3 Nonholonomic wheeled mobile robot . . . 3
1.4 a) Caster wheel, b) Conventional wheel, c)Omnidirectional wheel [22] and d) Ball wheel [22] . . . 4
1.5 A mecanum wheel mobile robot platform [28] . . . 5
1.6 Using PCW in different WMR configurations: a) 4-wheeled platform configuration [32], b) Stanford University PCW [38] c) 3-wheeled platform configuration[36] . . . 6
1.7 RNBC control structure . . . 10
1.8 C3P platform construction . . . 12

2.1 C3P Configuration . . . 19
2.2 C3P Coordinates Conventions . . . 20
2.3 The C3P parts structure . . . 32
2.4 C3P Dynamic model . . . 36

3.1 Different steering configurations . . . 38
3.2 Coupling between $\dot{\theta}_{x_1}$ and $\dot{\theta}_{x_2}$ . . . 39
3.3 Open loop structure using C3P dynamic model . . . 43
3.4 C3P wheel configuration considering the results in Figure (3.5) . . . 44
3.5 The C3P simulation results from example (1); driving in $\dot{\mathbf{p}} = [0.12(m/s)\ 0.12(m/s)\ 0(r/min)]^T$ . . . 45
3.6 C3P wheels configuration considering the results in Figure (3.7) . . . 46
3.7 The C3P simulation results from example (2); driving in $\dot{\mathbf{p}} = [0(m/s)\ -0.12(m/s)\ 0(r/min)]^T$ . . . 47
3.8 The geometric representation for achieving Ψ . . . 48
3.9 C3P simulation results with and without WCE regulator . . . 49
3.10 C3P velocity controller structure . . . 51
3.11 C3P velocity controller structure . . . 51
3.12 C3P performance with and without the velocity controller . . . 52

3.13 Robot Position representation 54
3.14 Robot trajectory for $K_\phi = 20$ and different K_{er} 55
3.15 Robot trajectory for $\delta_{er} = 5$ and differentΦ_e 55
3.16 Robot Position update 56
3.17 Simulation result for driving from $\mathbf{p}_i = [0\ 0\ 0^o]^T$ to $\mathbf{p}_g = [3\ 3\ -90^o]^T$ 57
3.18 Position update and trajectory for $\mathbf{p}_g = (3\ 3\ -90^o)^T$ 58

4.1 Dynamics Based Velocity Control Structure 62
4.2 Position Control Structure 62
4.3 The Steering angles orientation for driving in X direction from configuration (a) to configuration (b) 64
4.4 The wheels velocities and acceleration for ramp input and driving from singular condition 65
4.5 The robot velocities and acceleration for ramp input and driving from singular condition 66
4.6 Comparing dynamic and kinematic inverse solutions for deriving in x direction from initial singular condition 67
4.7 The steering angles orientation for driving in y direction from configuration (a) to configuration (b) 68
4.8 Comparing dynamic and kinematic inverse solutions for mobility in y direction from an initial singular condition 69
4.9 Dynamics and Kinematics fusion block 70
4.10 Simulation results using Dynamics and Kinematics fusion 71
4.11 Robot Position representation. 72
4.12 Position controller results for driving between $\mathbf{p}_i = [0m, 0m, 0^o]^T$ to $\mathbf{p}_g = [-4.5m, -4.5m, 90^o]^T$. 74
4.13 The effect of the parameters K_ϕ and K_{er} on the robot trajectory. . 75
4.14 the effect of initial steering angles values on the C3P trajectories for two different examples. 76

5.1 The C3P open control structure 79
5.2 The values of $\dot{V}$ for equal uniform values of steering angles 83
5.3 The values of $\dot{V}$ for non-equal uniform values of steering angles . . 85
5.4 3-D space $\dot{V}$ representation for random values θ_{s_1} and θ_{s_3}, unstable: *, stable: * 87
5.5 2-D Lyapunov space with random θ_{s_1} and θ_{s_3} values for different K_x values at $\theta_{s_2} = 205^o$ values, a)K_x=0.5, b) K_x=0.9, c)K_x=1.1, d)K_x=1.5, unstable: *, stable: * 88

6.1 The C3P practical prototype 89

6.2 The lower level of the caster wheels units . . . 90
6.3 The slip rings . . . 90
6.4 The absolute encoder . . . 91
6.5 Velocity control cards . . . 91
6.6 Kinematics Based Control Structure . . . 92
6.7 The Steering angles orientation for driving in X direction from configuration (a) to configuration (b) . . . 92
6.8 C3P practical results for driving in x direction with open loop WCE 93
6.9 C3P results for driving in x direction with closed loop velocity control 95
6.10 C3P results for driving in y direction . . . 96
6.11 The Steering angles orientation for driving in -Y direction from configuration (a) to configuration (b) . . . 97
6.12 Position control experiment, driving from $\mathbf{p}_i = [0m\ 0m\ 0^o]^T$ to $\mathbf{p}_g = [-3m\ -3m\ 0^o]^T$ . . . 98
6.13 C3P trajectories and steering angles values for two experiments; a) driving to $\mathbf{p}_g = [0m\ -3m\ 0^o]^T$ and b) $\mathbf{p}_g = [4.2m\ 0m\ 0^o]^T$ . . . 99
6.14 The C3P practical velocity control loop structure . . . 100
6.15 The C3P hardware block . . . 100
6.16 The C3P practical results from the inverse dynamic and kinematic solutions for driving in x direction . . . 101
6.17 The C3P practical results from the dynamics based controller for driving in y direction . . . 103
6.18 The C3P practical results from the dynamics based controller for driving in (x,y) direction . . . 104
6.19 The C3P practical results for driving in ∞ shape . . . 105
6.20 The position controller behavior for driving from $\mathbf{p}_i = [0\ 0\ 0]^T$ to $\mathbf{p}_g = [4m\ 0m\ 0^o]^T$ . . . 106
6.21 The position controller behavior for driving from from $\mathbf{p}_i = [0\ 0\ 0]^T$ to $\mathbf{p}_g = [0m\ 4m\ 0^o]^T$ . . . 108
6.22 Comparing the Gyro & Odometry data to the Krypton data . . . 109
6.23 Comparing the C3P trajectory measured by the Krepton and the Gyro sensors . . . 111
6.24 C3P behavior for driving in triangle shape . . . 112
6.25 The C3P behavior for sudden disturbances in single steering angle . 113
6.26 The C3P behavior for sudden disturbances in the three steering angle . . . 115

7.1 HCWR Configuration Structure . . . 118
7.2 Omni Configuration Structure . . . 118
7.3 RAMSIS-II Configuration Structure . . . 119

7.4 Velocity control loop structure 120
7.5 Energies and Velocities for Driving in 3DOF 122
7.6 Robots' Trajectories for Driving in 3DOF 123
7.7 Energies and Velocities for Driving in (∞) Shape 124
7.8 Robots Trajectories for Driving in (∞) Shape 125
7.9 Evaluating the robots cost values for different weights for 3DOF . . 130
7.10 Evaluating the robots cost values for different weights for (∞) Shape 132

List of Tables

3.1 The C3P parameters 44
3.2 The controller xondition 51

4.1 The C3P Control parameters 63

5.1 The steering angles values combination 86

6.1 The C3P parameters 92
6.2 The position performance errors and evaluation 110
6.3 The states values at high disturbances for Figure 6.25 114
6.4 The states values at high disturbances for Figure 6.26 114

7.1 Robots Hardware Complexity Value 128
7.2 Robots Cost Values for Driving in 3DOF 131
7.3 Robots Cost Values for Driving in (∞) Shape 133

List of Symbols

$\mathbf{p}$	Position Vector, p. 18
${}^{A}H_{B}$	Transformation Matrix between co-ordinates B and A, p. 18
${}^{A}\theta_{B}$	Rotated angle between co-ordinates B and A, p. 18
$\dot{\mathbf{p}}$	Robot velocities vector, p. 21
$\dot{\mathbf{q}}$	The wheel velocities vector, p. 21
$\dot{J}_i$	The Jacobian matrix for the wheel number i, p. 21
x	Displacement along the X axis, p. 21
y	Displacement along the Y axis, p. 21
ϕ	Rotation angle around the Z axis, p. 21
d_{s_x}	The distance between the robot center of gravity and the steering Z axis in X direction, p. 21
d_{s_y}	The distance between the robot center of gravity and the steering Z axis in Y direction, p. 21
d_{y}	The distance between the robot center of gravity and the contact point Z axis in X direction, p. 21
d_{x}	The distance between the robot center of gravity and the contact point Y axis in X direction, p. 21
d	The length of the offset link, p. 21

r The radius of the wheel, p. 21

h The distance between the robot and the hip coordinates systems, p. 21

α_i The shifting angle between each caster wheel, p. 21

θ_{s_i} The angular distance between the robot and the hip coordinates systems, p. 21

$\dot{\theta}_{s_i}$ The steering joint angular velocity, p. 21

$\dot{\theta}_{x_i}$ The wheel angular velocity, p. 21

$\dot{\theta}_{c_i}$ The contact angular velocity , p. 21

$\dot{\mathbf{q}}_a$ The actuated wheels velocities, p. 22

$\dot{\mathbf{q}}_n$ The non-actuated wheels velocities, p. 22

$\dot{J}_{a_i}$ The actuated Jacobian matrix for the wheel number i, p. 23

$\dot{J}_{n_i}$ The non-actuated Jacobian matrix for the wheel number i, p. 23

J_{in_x} The angular wheel velocities inverse actuated Jacobian, p. 23

J_{in_s} The steering angularvelocities inverse actuated Jacobian, p. 23

J_{in_c} The contact angular velocities inverse actuated Jacobian, p. 23

$\mathbf{q}_x$, $\dot{\mathbf{q}}_x$, $\ddot{\mathbf{q}}_x$ The wheel angles, angular velocities and accelerations vectors, p. 24

$\mathbf{q}_s$, $\dot{\mathbf{q}}_s$, $\ddot{\mathbf{q}}_s$ The steering angles, angular velocities and accelerations vectors, p. 24

$\mathbf{q}_c$, $\dot{\mathbf{q}}_c$, $\ddot{\mathbf{q}}_c$ The contact angles, angular velocities and accelerations vectors, p. 24

$\dot{\mathbf{q}}_{sen}$ The sensed wheels velocities, p. 24

$\dot{\mathbf{q}}_u$ The un-sensed wheels velocities, p. 24

J_{s_i} The sensed Jacobian matrix for the wheel number i, p. 24

J_{u_i} The un-sensed Jacobian matrix for the wheel number i, p. 24

J_{f_x} The wheels velocities forward Jacobian, p. 24

J_{f_s} The steering velocities forward Jacobian, p. 24

$\mathbf{q}_g$ The generalized coordinates vector, p. 25

$B_n(\mathbf{q}_g)$ Is mxn dimensional matrix containing the coefficents of the generalized coordinates with respect to the constraint equations , p. 25

$M(\mathbf{q}_g)$ The nxn dimensional positive definite inertia matrix, p. 26

$G(\mathbf{q}_g, \dot{\mathbf{q}}_g)$ The n-dimensional velocity-dependent force vector, p. 26

λ_n The Lagrangian Multiplier for non-holonomic constraints, p. 26

τ The a_r-dimensional vector of actuator force/torque, p. 26

$E(\mathbf{q}_g)$ the nxa_r dimensional matrix mapping the actuator space into the generalized coordinate, p. 26

$B_h(\mathbf{q}_g)$ The Holonomic constraints, p. 26

λ_h The Lagrangian Multiplier of holonomic constraints equation, p. 26

$B_J(\mathbf{q}_g)$ The Jacobian of the holonomic constraints, p. 26

Δ_r The distribution spanned by null space of $B_n(\mathbf{q}_g)$, p. 27

Δ_r^* The smallest involutive distribution containing Δ_r, p. 27

$S(\mathbf{q})$	Spans the null space of $B_J(\mathbf{q}_g)$, p. 29
K_p	The platform kinetic energy equation, p. 33
m_p, I_p	The platform mass and inertia, p. 33
$\boldsymbol{V}_p$	The robot translational velocities vector, p. 33
Ω_p	The robot rotational velocities, p. 33
τ_x	The input torques vector to the Wheels Dynamics Equation, p. 33
M_x	A square matrix, which contains the mass and inertia parameters related to wheels angular accelerations $\ddot{q}_x$ of the Dynamics Steering Estimator, p. 34
M_s	A square matrix, which contains the parameters related to the steering angular accelerations $\ddot{q}_s$ of the Dynamics Steering Estimator, p. 34
τ_s	The Steering torques vector, p. 34
$G_{sx}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s)$	The centripetal and Cloris forces of the Dynamics Steering Estimator equation, p. 34
J_{in_s}	The steering velocities inverse actuated Jacobian, p. 35
τ_{p_l}	The robot forces/torque vector, p. 35
M_{p_l}	The robot mass inertia matrix, p. 35
J_{f_x}	The wheels velocities forward Jacobian, p. 35
J_{f_s}	The steering velocities forward Jacobian, p. 35
Ψ	The singularity indicator, p. 48
$\dot{X}_a, \dot{Y}_a$	The measured robot Linear velocities in X and Y directions, p. 48
$\dot{X}_r, \dot{Y}_r$	The reference robot Linear velocities in X and Y directions, p. 48

K_{P_x},K_{I_x},K_{D_x} The PID controller parameters for robot velocity control in X direction, p. 51

K_{P_y},K_{I_y},K_{D_y} The PID controller parameters for for robot velocity control in Y direction, p. 51

K_{P_Φ},K_{I_Φ},K_{D_Φ} The PID controller parameters for for robot rotational velocity control around Z axis, p. 51

K_x,K_y,K_Φ,K_{er} The position control parameters, p. 72

τ_{x_a} The actuated torques vector resulted from the inverse dynamics solution, p. 60

M_{xa} The mass matrix constrained by the wheel angular acceleration vector $\ddot{q}_x$ for the inverse dynamics solution, p. 143

M_{sa} The mass matrix constrained by the steering angular acceleration vector $\ddot{q}_s$ for the inverse dynamics solution, p. 143

G_{sx_a} The centripetal and Coriolis torques for the inverse dynamics solution, p. 60

J_{in_c} The contact velocities inverse actuated Jacobian, p. 23

J_{f_i} The forward direct solution for the wheel accelerations, p. 60

$J_{r_i}, \dot{\mathbf{q}}_{r_i}$ The centripetal and Cloris forces, p. 60

$\Gamma_{V.C}(S)$ the velocity control matrix used for the inverse dynamics solution, p. 63

K_{px}, K_{dx}, K_{ix} The PID control parameters for controlling the X direction velocity, p. 63

K_{py}, K_{dy}, K_{iy} The PID control parameters for controlling the Y direction velocity, p. 63

$K_{p\phi}$, $K_{d\phi}$, $K_{i\phi}$ The PID control parameters for controlling angular velocity around the Z axis, p. 63

$\mathbf{q}_i$	The initial position co-ordinates, p. 72
$\mathbf{q}_g$	The goal position co-ordinates, p. 72
τ_D	The wheels torques resulted from the inverse dynamic solution, p. 70
τ_K	The torque resulting from the wheel velocity axes control, p. 70
λ_1, λ_2	The fusion parameters , p. 70
r_o	The omnidirectional wheel roller radius, p. 155
R	The omnidirectional wheel radius, p. 155
$V(t)$	The Lyapunov function, p. 81
$\rho_c(t)$	The omnidirectional wheel radius, p. 80
$\dot{\theta}_{p_i}$	The roller angular velocity around the z axis, p. 155
$\dot{\theta}_{xr_i}$	The wheel angular velocity for Ramsis II, p. 158
$\dot{\theta}_{zr_i}$	The contact angular velocity for Ramsis II, p. 158
$\dot{\theta}_s$	The turret angular velocity for Ramsis II, p. 158
P_i	The power consumed by the actuator i, p. 120
P_T	The power consumed by the robot actuators, p. 120
E_T	Total energies consumed by the robot actuators, p. 121
C_{m_l}	Translational robot velocity cost function, p. 126
C_{m_r}	Rotational robot velocity cost function, p. 126
C_e	The robot Consumed energy cost functional, p. 126
C_k	Hardware complexity cost function for each robot, p. 129
J_r	The overall cost functional for each robot, p. 129

M_{ax_l}, M_{ax_r}, M_{ax_k}, and M_{ax_e}	The maximum cost values for the linear velocities error, rotational velocity error, hardware complexity, and energy consumption, p. 129
w_1, w_2, w_3	The WMR cost functional weights for the mobility, hardware complexity, and energy consumption, p. 129
μ_x,μ_y	Position Control parameters, p. 72

Chapter 1

Introduction

1.1 Overview

Mobile robots are widely integrated in our present society in many public places such as shopping centers [1] and airports [2]. Therefore, over the last decades the field of mobile robots has encountered several challenges in a considerable number of researches. *Mobile robot* is a collection of algorithms for sensing, reasoning, and moving about space, in addition to the physical embodiments of these algorithms and ideas that must cope with all the vagaries of the real world [3]. The mobile platforms can be divided into two main categories: legged and wheeled platforms. Our work in this thesis concentrates on the development of the wheeled mobile robot. The *Wheeled Mobile Robot (WMR)* is a robot capable of mobility on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface [4]. *Wheel assembly* is a device that provides or allows relative motion between its mount and a surface on which it is intended to have a single point of rolling contact.

Wheeled mobile robots are found in a host of applications such as guiding disabled people in museums [5][6][7] and hospitals [8], transporting goods in warehouses, manoeuvring army explosives (Fig.1.1) [9], or securing important facilities [10][11].

Wheeled mobile robots are categorized in two main types: *holonomic* and *non-holonomic*, which are the mobility constraints of the mobile robot platform [12]. A holonomic configuration implies that the numbers of robot velocity DOF (Degrees

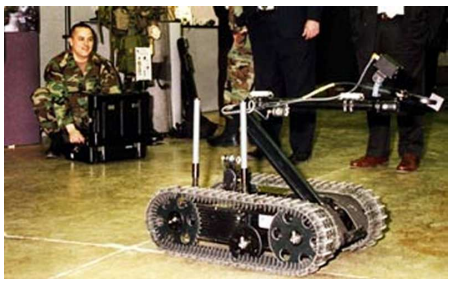

Figure 1.1. Using WMR as explosives transporter [9]

Of Freedom) are equal to the number of position DOF. For example, a rigid body has six degrees of freedoms, which are the position on the three dimension axes X, Y and Z and the rotational angles around each axis (Fig.1.2) .

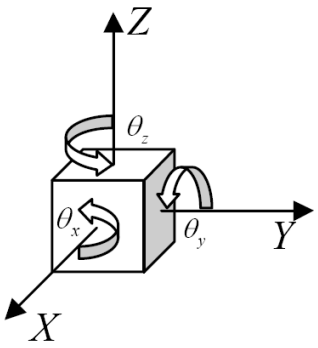

Figure 1.2. Rigid body degrees of freedoms

The WMR normally moves on a planner surface with three position coordinates: X, Y and rotational angle around Z which is θ_z. Therefore, the holonomic WMR is the robot that can drive in three degrees of freedom (3DOF), and the nonholonomic WMR is the robot that can not perform 3DOF mobility. The nonholonomic WMR can not move sideways, as shown in Figure 1.3.

1.2 Motivation

Over the last decades, wheeled mobile robots (WMR) have attracted the attention of many researchers. WMRs are developing rapidly, along with their hardware and software structures to achieve their main goals [16][17][18][19]. One of the

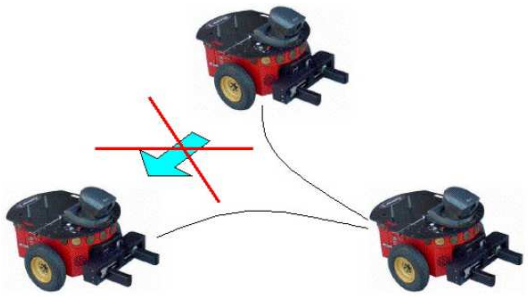

Figure 1.3. Nonholonomic wheeled mobile robot

important goals is to solve problems affecting the robot mobility behaviour. As a result, many complex platforms were developed to achieve 3DOF mobility (i.e to avoid nonholonomic characteristics). Such platforms are usually equipped with a complex wheel set-up (e.g: two sided roller wheels, complex special geared castor wheels or ball wheels). The main disadvantages of the mentioned design variants are the high energy consumed by the actuators and the required frequent maintenance. That is why WMRs became one of the complex engineering systems to be designed [20][21]. So far, reducing such complexity has not received much attention because some robots are experimental prototypes which are not exposed to the rigorous demands of commercial products. Thus, the main goal of this work is to deliver a WMR platform with the minimum possible number of components without alerting the holonomic features of the platform.

A simple configuration for holonomic WMR platform in [13] is achieved by reducing the number of actuators and choosing a suitable 3DOF wheel set-up to maintain the holonomic characteristics. Such a configuration resulted in singularity issues that affects the robot mobility behaviour. The main theoretical challenge is using both kinematics and dynamics modeling to obtain a singularity free solution, along with its motion controller. Due to the reduction of actuators, the system non-linearities will increase and investigating the system stability will become more challengable.

To show the advantages of such a platform configuration, it will be compared with other holonomic WMRs using a special noval criteria for evaluating WMRs. So far, no basic quantitative method has been found in literature for evaluating different WMRs. Few researches have tackled this problem from a single aspect point

of view [20][21]. On the other hand, there are many aspects affecting the WMR evaluation, for example: mobility behaviour, platform construction, hardware set-up, electrical set-up, software design, and energy consumption. The challenge is to obtain a cost evaluation method, which delivers a measured quantity for the WMR with respect to its main affective aspects.

1.3 State of the Art

In the last two decades, a number of considerable research efforts addressing the mobility of holonomic wheeled mobile robots have been carried out [22][23][24]. The mobility behavior of the WMR depends mainly on the type of its wheels and their actuated velocities. There are four main types of wheels used in the WMR: a) caster wheel (Fig. 1.4a), b) conventional wheel (Fig. 1.4b), c) omnidirectional wheel (Fig. 1.4c) [22], and d) ball wheel (Fig. 1.4d) .

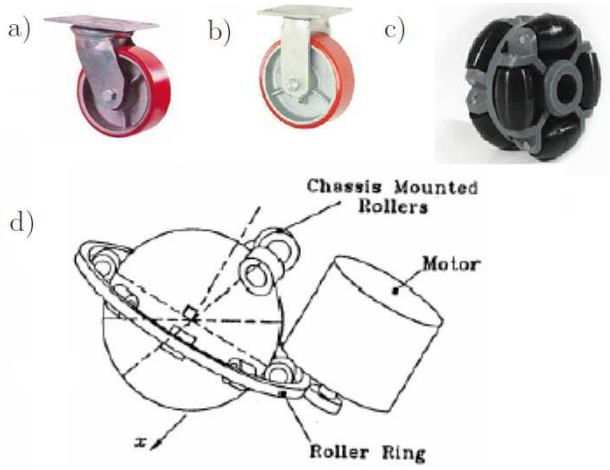

Figure 1.4. a) Caster wheel, b) Conventional wheel, c)Omnidirectional wheel [22] and d) Ball wheel [22]

The Omnidirectional Wheel construction consists of rollers mounted around the main wheel. The wheel motion depends on the angle between the roller axis and the wheel rotating axes as shown in Figure (1.4d) [25][26]. The conventional wheel is the only wheel that has 2DOFs mobility and is the simplest in construction. The conventional wheel is used mainly for the non-holonomic WMRs, along with

the caster wheel to support the platform balance. The Omnidirectional Wheel has 3DOF's mobility [15], therefore it is normally used in the holonomic WMRs configurations, as shown in Figure (1.5).

In this section holonomic WMR platforms are briefly discussed to give an overview of the state of the art in the holonomic WMRs. Some WMRs use ball wheels to achieve holonomic mobility, such as the robot *Cobot*[27]. The Cobot has three ball wheels with powerful actuation control and synchronization. A roller drive system is mounted above each wheel. The system consists of a sphere actuated by 6 rollers, where by each pair of rollers is used for one degree of freedom actuation. This kind of actuation requires a very complex mechanical structure and high energy consumption in addition to regular intensive maintenance.

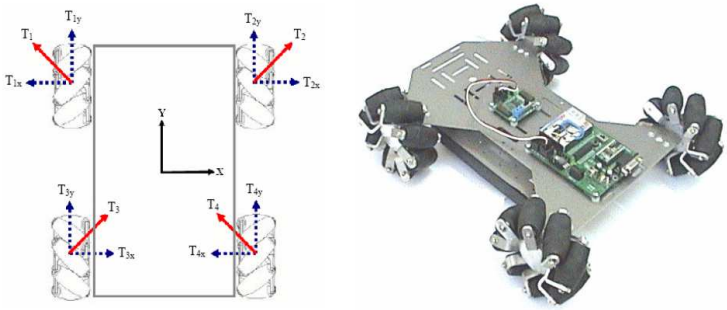

Figure 1.5. A mecanum wheel mobile robot platform [28]

The Omnidirectional Wheels show their efficient performance with different WMR platforms. For example, in [28] a design of a holonomic mobile robot with omnidirectional wheeled designed by Mecanum AB' Bengt Ilon, shown in Figure (1.5). The mecanum wheel developed consists of nine rollers made from delrin. Typical mecanum wheel mobile robot platforms are square or rectangular, attached with a wheel with a +45 roller and a wheel with -45 roller on each side. The omnidirectional capabilities of the platform depend on each wheel contact having firm contact with the surface, where some of the mecanum wheel mobile robots are equipped with a suspension system (Fig.1.5). The main disadvantage of the omnidirectional wheel is its complex construction and its difficult complicated maintenance.

The caster wheel has proved its efficient performance in 3DOF mobility appli-

cations. It has different configurations which are used in normal life applications [29][30]. Therefore the caster wheel is widely used in the WMR's platform configurations. The caster wheel can be used as a passive wheel (as in 2DOF platforms [31]) or an active wheel as in [32]. Many WMRs with Powered Caster Wheels (PCWs, also known as offset steerable wheels) have been developed [33][34][35] and even commercialized. One major benefit of using PCWs is that WMRs with PCWs can generate 3DOF mobility.

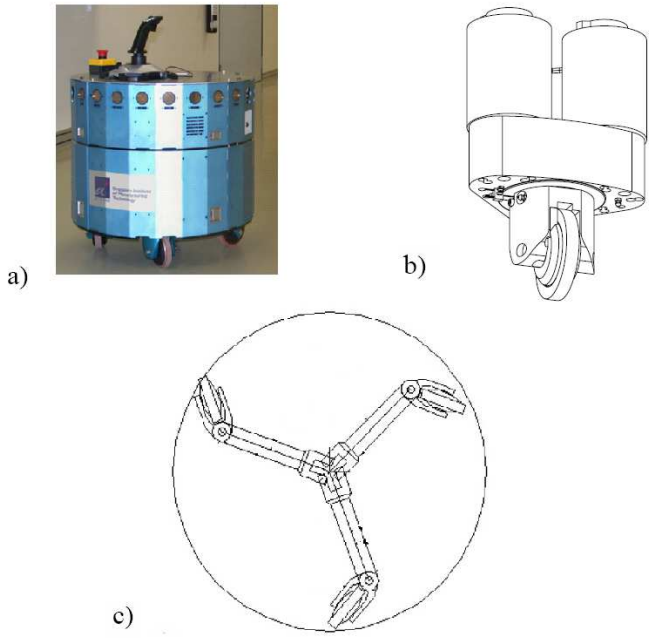

Figure 1.6. Using PCW in different WMR configurations: a) 4-wheeled platform configuration [32], b) Stanford University PCW [38] c) 3-wheeled platform configuration[36]

In [32] the platform has four PCWs and configured in the following manner; PCWs have steering and rolling angular velocities actuation (Fig.1.6-a). Similar actuation configuration is also presented in [36], but with three PCWs (Fig.1.6-c). The actuation of the steering and the rolling axes depends mainly on the mechanical structure of the PCW. The robotics team at Stanford University (USA) developed a powered caster wheel system configuration [38] (Fig.1.6-b), which assigns a complex mechanical gear unit for each wheel to actuate the steering of the wheels. Such a mechanism requires high energy and power consumption, complex dynamic control, and large wheel radius. In [39] the authors used a simplified PCW

configuration to actuate the driven angular velocity of two caster wheels and the steering axes of the third wheel. This showed interesting simulation results, but for a practical application the steering actuation still requires high energy to produce enough torque to adjust the wheels' steering angles in the desired direction.

1.3.1 Kinematics Modeling

The WMR is a *multibody* system, which is defined as an assembly of two or more rigid bodies (also called elements) imperfectly joined together, having the possibility to relative movement between each other. This imperfect joining of two rigid bodies that makes up a multibody system in called a *Kinematic pair* or *joint* [40].

Kinematic problems are those in which the position or motion of the multibody system are studied. The kinematic modeling has pure geometrical nature with no repect to the dynamic parameters such as mass, inertia and friction. Usually, kinematic modeling is used in the field of WMRs to obtain stable motion control laws for trajectory following or goal reaching [41] [42]. The kinematic modeling method and analysis of mobile robots equipped with the previously mentioned types of wheels were proposed [43] [44]. In [4] the kinematic modeling of the robots is directly performed in the motion space. Untill now, the methods suggested in [4] have been widely used in kinematic modeling of various types of wheeled mobile robots. Due to the relative simplicity and high effectiveness of the kinematic model, it is the first step in building a wheeled mobile robot under the following basic assumptions: the floor is stationary and planar, no wheel-slip in the direction of translation, a rotational slip is necessary around the steering axis, there is no elasticity in any part, maximum one steering link per wheel, and the steering-axis perpendicular to floor. The violation of any of these assumptions will lead to the modelling of the slip and compliance, which is out of the scope of the work.

1.3.2 Dynamic Modeling

The Dynamic model is more difficult to derive than the kinematic model. The kinematic modeling is required for deriving the dynamic model. Hence, it is to be assumed that the velocity and acceleration solutions can be easily obtained. Generally, the main property of the dynamic model is that it involves the forces

that act on the multibody system and its inertial parameters, such as : mass, inertia, and the center of gravity.

the Dynamic modeling consists of two main solutions: the forward dynamic solution and the inverse dynamic solution. The *forward dynamic* solution yields the motion of a multibody system over a the given time interval, as a consequence of the applied forces and given initial conditions. The importance of the direct dynamic model lies in the fact that it allows the simulation and prediction of the system's actual behavior; motion is always the result of the forces that produce it. The *inverse dynamic* solution aims at determining the motor or the driving forces that produce a specific motion, as well as the reactions that appear at each joint of the multibody system [45].

In this work we will be primarily interested in robots consisting of a collection of rigid links connected through joints that constrain the relative motion between the links. The dynamic modeling has been studied by several researches for many mobile robot platforms [46] [47] [48] [49].

Two main methods exist for deriving the dynamic equations for such mechanical systems: a) Euler-Lagrange and b) Newton-Euler formulations[50] [51] [52]. The main difference between the two approaches is how they deal with the system constraints. Newton-Euler equations are directly based on Newton's laws, which treat each rigid body separately and explicitly, and model the constraints through the forces required to enforce them. Newton-Euler approaches use Cartesian variables as configuration-space variables, they admit recursive formulations by first developing the equation of motion for each single body; these equations are then assembled to obtain the model of the entire system [53][54]. Euler-Lagrange formulations use joint-based relative coordinates as configuration-space variables; these formulations are generally not well suited for recursive formulation. Lagrange and d'Alembert provided systematic procedures for eliminating the constraints from the dynamic equations, typically yielding a simpler system of equations. Constraints imposed by joints and by other mechanical components are one of the defining features of robots so it is not surprising the Lagrange's formalism is often the method of choice in robotics literature [55] [56] [57] [58] [59] [60]..

1.3.3 Wheeled Mobile Robot Control Structure

The WMR is capable of an autonomous motion (without an external human driver) because it is equipped for its motion with drivers that are controlled by an embarked computer [61]. An individual autonomous mobile robot requires a reliable control structure, which achieves the following requirements [63]: a) treats conflicts in reaching *multiple goals*; b) maintaind *robustness*, in performance in general and stability in practicality, against ornamental uncertainties, sensor noise and actuator inaccuracy; c) allows *recursive realization* to provide natural extendability of the behavior from coarse and reflexive to fine and deliberative; d) posses *learning capabilities* preferably distributed and tailored to the knowledge representation; e) accommodates *sensor and actuator hierarchy.*

In [64][65] a so-called "Recursive Nested-Based Control" (Fig. 1.7) (RNBC) structure has been successfully employed for individual autonomous mobile robots. The main properties of the RNBC structure are:

- It can be viewed as a generalization of the cascaded control.
- The behavior levels are nested, which provides an inherent robustness against loop failure.
- It is recursive, where the interactions are done between the i^{th} and $(i+1)^{th}$ levels as well as between the i^{th} and $(i-1)^{th}$ levels if exists, however there is no interactions between the $(i+1)^{th}$ and $(i-1)^{th}$ levels.
- It is a bottom-up approach, providing a gradual increase in the control structure.
- No explicit sensor-fusion is necessary where the sensors provide their data to the level they are first needed and then they are fed to each other forward or backwards with consequent delay.
- Different behavior levels are used according to the level of abstraction where behavior fusion is performed using networks of analogical gates [65].

The RNBC structure is shown in Figure (1.7) with 8 levels of control, based on the dynamic and kinematic models of the WMR. The axis-level control is the classical control loop of each actuator that produces its torque actuated signal. The

robot control is established in level two for controlling the WMR as a whole. After controlling the WMR velocities a controller is used in level three to avoid colliding with obstacles in the way of the robot. Homing is basically the position control loop that drives the WMR from one point to another without any trajectory planning. Level five contains the method that is used in updating the robot position for its local navigation in level six. The local navigation builds a line of sight between the robot goal and its current position, if the goal is invisible sub-goals

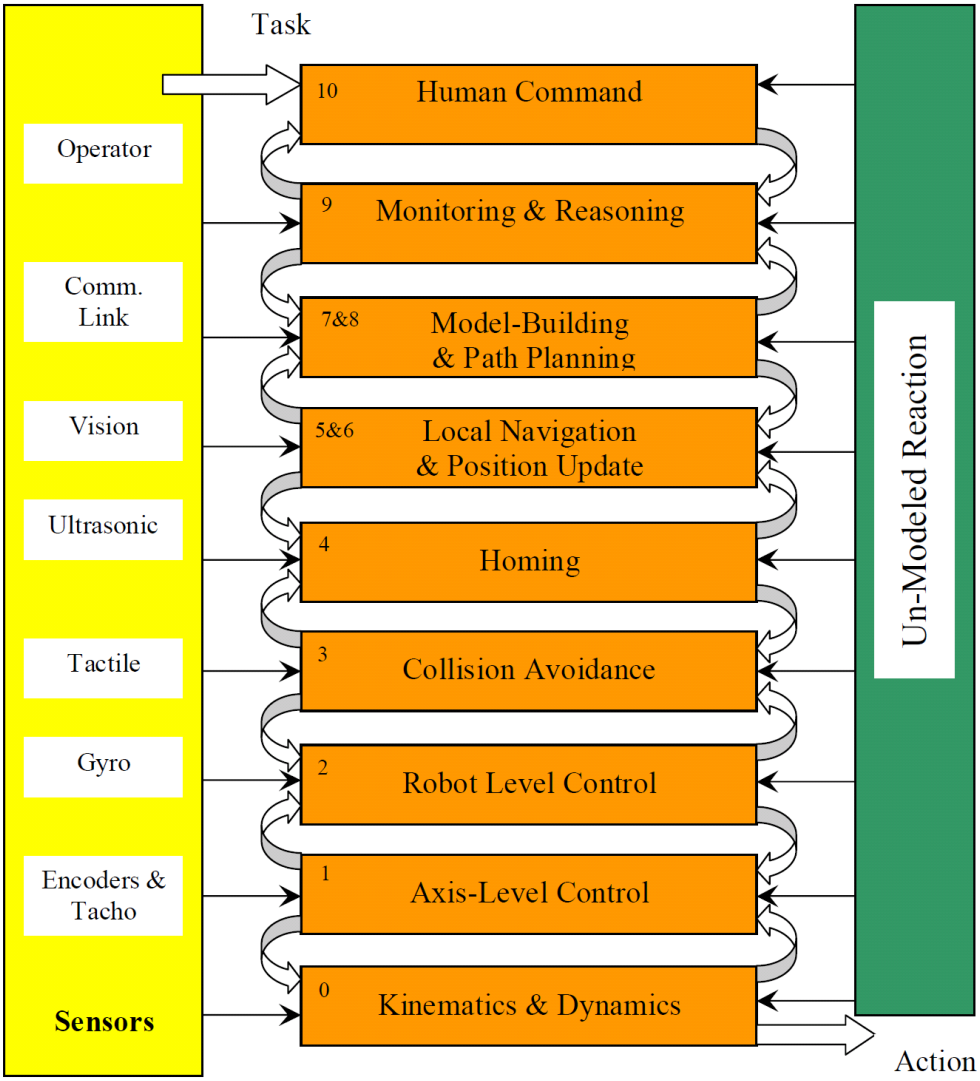

Figure 1.7. RNBC control structure

are created to track such a line. The level of path planning and model building requires the robot to define an optimal path to reach a certain goal, such a path is defined based on a map model built by the robot depending on the surrounding environment. Each level has its own local monitor to prevent reaching some forbidden conditions for its states. The monitoring and reasoning level incorporates the monitors of the whole robot. The last level is the human command, which enables the operator to command the robot according to its last implemented level, for example a joy-stick is used if the robot control is the last level. In this thesis, the work mainly concentrates on just a few levels, such as Kinematic and Dynamic Modeling, Robot Level Control, Position Update and Trajectory control. Generally, modeling and motion control problems are tackled in this work. There are two main motion control tasks for WMRs, stabilizing to an equilibrium point (such as parking) and stabilizing to an equilibrium manifold (such as trajectory tracking or path following) [31]. The first control task is considered challenging because WMR's with different configurations cannot be stabilized to an equilibrium point [66] [67]. The second problem is the stabilization to an equilibrium manifold or the trajectory control. The trajectory controller or tracker depends on the inverse dynamic solution, which helps by providing smooth and successful maneuvering.

1.4 Problem Formulation

The work of this thesis is inspired by the main objective of 'Building a holonomic wheeled mobile robot that is simple, modular and efficient in its performance'. This objective is reached by three main points. Firstly, the WMR should be assembled of the simplest constructed 3DOFs wheel type with low maintenance requirements. Secondly, the WMR's actuated velocities should be modular, easily actuated and minimum in their numbers. Thirdly, the kinematic and dynamic analyzing and modeling should be obtained to employ effective robot velocity, position and trajectory controllers.

As a conclusion to Section 1.1, the caster wheel is the simplest 3DOF constructed wheel with minimum requirements for maintenance as well. Using three wheels for a WMR has the advantage that wheel-to-ground contact can be maintained on all wheels without a suspension system [68]. Therefore, three caster

wheels are used in the construction of the holonomic WMR proposed in this thesis. Figure 1.8 shows the proposed platform, which is called 'C3P' (**C**aster **3** Wheeled **P**latform).

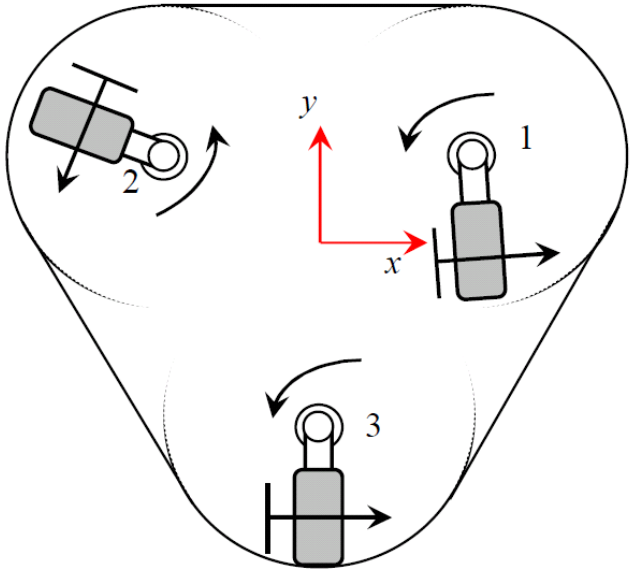

Figure 1.8. C3P platform construction

The platform configuration was proposed and discussed in [13]. The actuated velocities are modular for the following reasons: a) similar axis level controllers are used, b) the energy consumed is redistributed to the actuators and c) in the event an actuator has failure blocking, neither the used backup model nor its controller will change. Therefore, each caster wheel is actuated by its driven velocity (actuated angular velocity $\dot{\theta}_x$). The value of each caster wheel steering angle (θ_s) is needed in the kinematic and dynamic modeling;, therefore it is the main sensed element, along with the driven velocity ($\dot{\theta}_x$).

The C3P has a singularity problem for some wheel steering configurations, described as the following: when all wheels yield the same steering angle value the C3P can be actuated in any direction parallel to the wheel angular velocity axis.

With the given C3P platform and its actuated/sensed elements, it is required in this thesis to deliver the following:

1. Actuation and sensing analysis.

2. Forward and inverse kinematic model.
3. A dynamic model for the simulation process.
4. Overcome the actuating and sensing problems, if they exist..
5. Designing velocity and position controllers for the kinematic solution.
6. An inverse dynamic model based on velocity and position control.
7. Studying the system stability and analyzing its non-linearities.
8. Applying the proposed solutions and controllers on a C3P practical prototype.
9. Comparing the robot performance and construction with other holonomic mobile robots.

1.5 Main Contributions

This thesis investigates the performance of the new actuation configuration of three caster wheeled mobile robots. The contribution of this thesis consists of:

1. The C3P configuration is kinematicaly modeled using the methods in [4]. The **W**heel **C**oupling **E**quation (WCE) is proposed to overcome the singularity problem by virtualy actuating the steering angular velocities [69]. A special structure velocity control structure is proposed, which consists of WCE regulator and robot PID velocity controller [70].

2. The C3P forward dynamic model is obtained using the Euler Lagrangian equation. The model consists of two main dynamic equations; the WTD (**W**heel **T**orques **D**ynamics), and the DSE (**D**ynamic **S**teering **E**stimator) [71].

3. An inverse dynamic solution is developed to create a more accurate and feasible solution to avoid the assumptions and approximations used in the inverse kinematic solution [72] [73]. The solution is used in the development of the C3P velocity and position controllers.

4. The C3P platform is built by the Automation Laboratory. The first prototype platform is used for the practical experiments to illustrate the performance of the proposed models, solutions and control structures [74].

5. A comparison is done between the C3P and three holonomic mobile robots; **H**olonomic **C**aster **W**heeled **R**obot (HCWR), **O**mni **D**irectional **W**heeled **R**obot (ODWR), and RAMSIS II. The comparison is done with respect to the energy consumed by the robot to drive in a specific direction during a finite interval of time, the trajectory error of the robot, and the robot output velocities [75].

1.6 Outline

The thesis consists of eight chapters devoted to delivering the main objective mentioned in Section 1.4. Chapter 1 states the state of the art, along with the problem formulation, and the rest of the chapters are organized as the following:

Chapter 2 shows the kinematic and the dynamic modeling of the C3P. Firstly, the kinematic inverse and forward solution is obtained. Secondly, the constraints of the platform are proven to be holonomic and integrable. Thirdly, by using the Euler Lagrangian principle, a forward dynamic model for the C3P is obtained for the simulation processes that are done under the Matlab environment.

Chapter 3 illustrates the C3P singularity problem, which has been found for some wheel configuration. The coupling wheel approach (WCE)is proposed to solve such a problem kinematically, which depends on the coupling action between each of the two wheel's angular velocities to actuate the third wheel steering angular velocity. A special structural velocity controller is developed for regulating the WCE and controlling the C3P velocities. Moreover, the position controller is also proposed, in addition to several simulation examples illustrating the performance on the dynamic model and inverse/forward kinematic solutions.

Chapter 4 yields an inverse dynamic solution based on the Euler Lagrangian principle. It is based on employing the C3P inverse kinematic solution for actuating the steering and the contact angular velocities with in the solution to overcome the singularity problem. Velocity and position controllers are developed with the inverse kinematic model. The simulation results demonstrate the controllers' performances.

Chapter 5 shows the system model stability analysis using the Lyapunov Direct method. The Lyapunov function is developed based on the robot energy equation, which resulted in a quadratic equation as a function of the robot position

coordinates. The robot position variables are the three main states, while the steering angles values are considered as the system disturbances.

Chapter 6 illustrates the hardware equipment used for building the C3P practical prototype. The experimental results for each solution and its controller demonstrate the performance on the C3P practical prototype. Moreover, the prototype problems are pointed out; for example, the platform misconstruction parameters, the slippage, friction, and sensor problems.

Chapter 7 verifies the importance of the C3P configuration among other holonomic WMRs. A comparison between four holonomic WMRs is established. The robots are the C3P, HCWR (Holonomic Caster Wheeled Robot), The Omnidirectional Wheeled Mobile Robot and Ramsis II. The comparison is done on the simulation level to demonstrate the differences between each robot performance. The simulation shows the mobility and energy consumptions performance, in addition to hardware complexity. Cost function is obtained using the weighted sum method to evaluate the cheapest platform. The cost function is based on equal weighting of three main aspects; mobility error, energy consumption, and hardware complexity.

Chapter 8 illustrates the conclusion of the work in this thesis , in addition to suggested solutions for the first practical C3P prototype platform.

Chapter 2

The C3P Kinematic and Dynamic Modeling

2.1 Introduction

The kinematic model is the first step for WMR mobility analysis and control. The position, velocity, and acceleration constraints are determined according to the WMR wheel's configuration. Typical types of wheels used for WMRs can be classified as conventional, omnidirectional, ball, and caster wheels [43]. The last two kinds of wheels are kinematically modeled as a 3DOF serial chain, while the conventional wheel is modeled as a 2DOF serial chain. Untill the writing of this thesis, the method suggested in [4] has been used in kinematic modeling for different WMR configurations. The work develops a formalism that is used first to model the kinematics of each wheel, and second to amalgamate the information about individual wheels to describe the kinematics of WMR regarded as a whole. Generally, this method does not incorporate the friction model, such as sliding and skidding velocities into its kinematic model.

The structure of a WMR is a parallel kinematic structure which consists of serial sub-chains (the wheels). The WMR kinematic model will be obtained under a few assumptions. First, the dynamics of the WMR flexible suspension mechanisms and tyres are negligible. Second, all steering axes are perpendicular to the surface

of travel. Third, the WMR drives on a horizontal planar surface.

In order to verify a proposed robot axis, velocity or position control structure, it is necessary to have an accurate model for the WMR. The dynamic model has both geometrical and physical characteristics [76][77], which makes it more accurate than the kinematic model. A number of methods to formulate the equation of motion, were developed [78] [55]. The Lagrangian formulation is used to model mobile robot dynamics, considering the robot as a multibody closed-chain system with constraints.

The kinematic modeling methods used in this chapter are obtained in [4]. Such methods are widely used by the wheeled mobile robots community [79][80][81] and [82] .

2.2 Kinematic Modeling

Robot mechanisms are modeled as a chain of several rigid bodies (links) connected by either revolute or prismatic joints driven by actuators. This chain can be an open loop system (fixed at one end and free at the other), for example, a robot arm manipulator or a closed loop system (fixed at both ends), for example, the wheeled mobile robot. The wheeled mobile robot kinematics deals with the analytical study of robot geometrical motion with respect to a fixed reference coordinate system as a function of time without regarding the torque/forces. Thus, it deals with the analytical description of the spatial displacement of the WMR as a function of time, in particular the relations between the joint-variable space and position and orientation of the WMR center of gravity.

As an overview, this section will represent a refreshment for the basic methods of kinemtic modeling for rigid body. Since we will be concerned with robots consisting of rigid links, we start by describing rigid body motion. Formally, a rigid body O is a subset of R^3 where each element in O corresponds to a point on the rigid body. The defining property of a rigid body is that the distance between two arbitrary points on the rigid body remains unchanged as the rigid body moves. If a body-fixed coordinate frame B is attached to O, an arbitrary point $\mathbf{p} \in O$ can

be described by a fixed vector ${}^{B}\mathbf{p}$. As a result, the position of any point on O is uniquely determined by the location of the frame B. To describe the location of B in space we choose a global coordinate frame A. The position and orientation of the frame B in the frame A is called the configuration of O and can be described by a 4x4 homogeneous matrix ${}^{A}H_{B}$.

$$ {}^{A}\mathbf{p} = {}^{A}H_{B} \;\; {}^{B}\mathbf{p} \tag{2.1} $$

The homogeneous matrix ${}^{A}H_{B}$ (4x4) contains two characteristics, the rotational relation ${}^{A}\Phi_{B}$ (3x3) and the translation relation ${}^{A}d_{B}$ (3x1) as shown

$$ {}^{A}H_{B} = \begin{bmatrix} {}^{A}\Phi_{B} & {}^{A}\mathbf{d}_{B} \\ 0 & 1 \end{bmatrix}, {}^{A}\Phi_{B} \in R^{3X3}, \; {}^{A}\mathbf{d}_{B} \in R^{3}, \; {}^{A}\Phi_{B}^{T}{}^{A}\Phi_{B} = I_{3} \tag{2.2} $$

The kinematic modeling coordinate system is assigned with the z-axes perpendicular to the planar surface, therefore all rotations between coordinate systems are about the z-axis. The homogeneous matrix in the WMR kinematic model is a rotation of ${}^{A}\theta_{B}$ about the z-axis of coordinate system of point A and the translation ${}^{A}d_{B_x}$, ${}^{A}d_{B_y}$ and ${}^{A}d_{B_z}$ along the respective coordinate axes:

$$ {}^{A}H_{B} = \begin{pmatrix} \cos({}^{A}\theta_{B}) & -\sin({}^{A}\theta_{B}) & 0 & {}^{A}d_{B_x} \\ \sin({}^{A}\theta_{B}) & \cos({}^{A}\theta_{B}) & 0 & {}^{A}d_{B_y} \\ 0 & 0 & 1 & {}^{A}d_{B_z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.3} $$

The variables of the homogeneous matrix ${}^{A}H_{B}$ are defined below

${}^{A}\theta_{B}$: the angle between frame B and A.

${}^{A}d_{B_i}$: the distance between frame B and A where, along the axes $i = x, \; y, \; and \; z$.

2.3 The C3P Kinematic Modeling

The C3P is a holonomic mobile robot with three caster wheels as shown in Figure 2.1. Each caster wheel is attached to each corner of the platform with a distance

of $a = 0.5$ m away from each other. The wheel's radius is $r = 0.04$ m and the caster wheel offset distance is $d = 0.04$ m. The origin of coordinates frame of the platform are located at its geometric center, the wheels are located away from the origin with distance $h = 0.343$ and $\alpha_1 = 30^o$, $\alpha_2 = 150^o$, and $\alpha_3 = 270^o$ shifting angles. The angle $\dot{\theta}_{s_i}$ is the steering anglular velocity for wheels 1, 2 and 3.

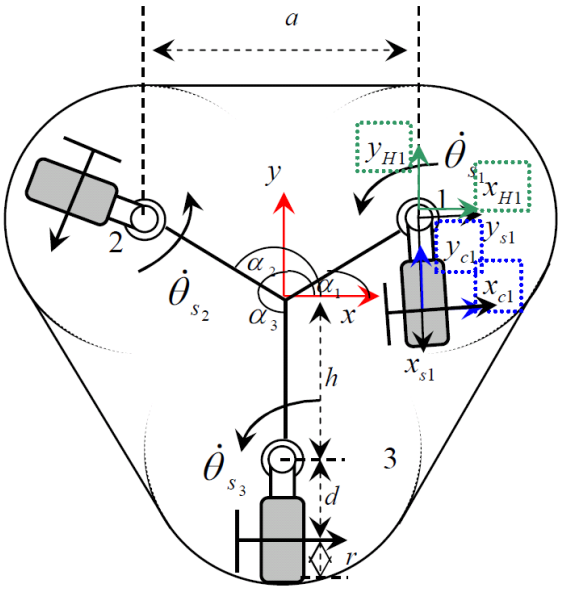

Figure 2.1. C3P Configuration

The main objective of this work is to deliver a WMR with holonomic mobility in 3DOFs. In order to achieve such mobility all the wheels attached to the platform should have 3DOFs mobility. In [4] a method was developed to obtain the robot velocities solution described by the Jacobian for the generalized wheel in Appendix A. This Jacobian is presented as follows:

$$\begin{bmatrix} {}^{\bar{B}}\dot{x}_B \\ {}^{\bar{B}}\dot{y}_B \\ {}^{\bar{B}}\dot{\phi}_B \end{bmatrix} = \begin{bmatrix} \cos({}^B\theta_C) & -\sin({}^B\theta_C) & {}^Bd_{C_y} & -{}^Bd_{H_y} \\ \sin({}^B\theta_C) & \cos({}^B\theta_C) & -{}^Bd_{C_x} & {}^Bd_{H_x} \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} {}^{\bar{C}}\dot{x}_C \\ {}^{\bar{C}}\dot{y}_C \\ {}^{\bar{C}}\dot{\theta}_C \\ {}^H\dot{\theta}_S \end{bmatrix} \tag{2.4}$$

where

F *Floor*: The stationary reference coordinate system, where the z-axis is orthogonal to

the planar surface.

B *Body* : The WMR body coordinate system.

H *Hip* : The coordinate system, which moves with the body for the steering joint.

S *Steering* : The steering coordinate system which moves with the steering link with z-axis coincident with the z-axis of the Hip.

C *Contact Point* : The contact point coordinate system .

$\overline{B}$ *Instantaneously Coincident Body* : The coordinate system Coincident with the B coordinate system relative to the stationary F coordinate system.

$\overline{C}$ *Instantaneously Coincident Contact Point* : The coordinate system Coincident with the C coordinate system relative to the stationary F coordinate system.

Figure 2.2 illustrates the coordinate frame used for the C3P mobile robot. The robot is given a body of refrence frame. This frame is usually at the geometric center of the robot. Each wheel is aslo given a frame.

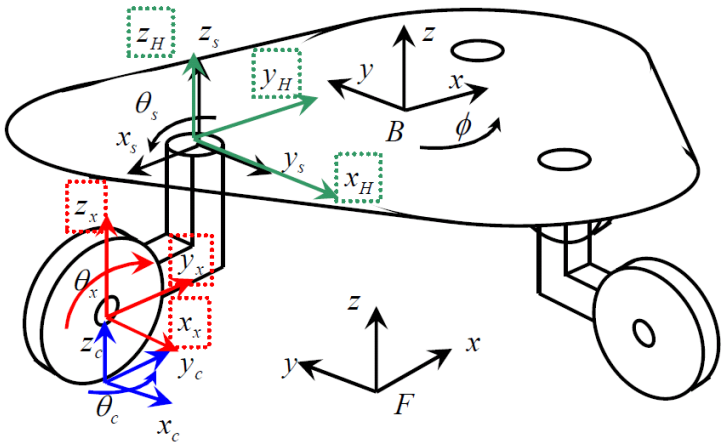

Figure 2.2. C3P Coordinates Conventions

For the C3P platform the caster wheel is used to achive such mobility for the reasons mentioned in Section (1.1). This section presents the kinematic model of the caster wheel, which is used in the platform construction. The wheel 3DOFs are provided by the steering joint angular velocity $\dot{\theta}_{s_i} = {}^{H}\dot{\theta}_{S}$, the wheel angular

velocity $\dot{\theta}_{x_i} = \frac{\bar{C}\dot{y}_C}{r}$, and the contact angular velocity $\dot{\theta}_{c_i} = {}^{\bar{C}}\dot{\theta}_C$ as shown in Figure 2.2. The kinematic relation between the robot velocities $\dot{\mathbf{p}}$ and the i^{th} wheel angular velocities vector $\dot{\mathbf{p}}_i$ is the Jacobian matrix J_i

$$\dot{\mathbf{p}} = J_i \ \dot{\mathbf{q}}_i$$
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -r\sin(\theta_{s_i}) & d_{y_i} & -d_{s_{yi}} \\ r\cos(\theta_{s_i}) & -d_{x_i} & d_{s_{xi}} \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{x_i} \\ \dot{\theta}_{c_i} \\ \dot{\theta}_{s_i} \end{bmatrix} \tag{2.5}$$

where

d_{s_x} : The distance between the robot center of gravity and the steering Z axis in X direction.

d_{s_y} : The distance between the robot center of gravity and the steering Z axis in Y direction.

d_{y} : The distance between the robot center of gravity and the contact point Z axis in X direction.

d_{x} : The distance between the robot center of gravity and the contact point Y axis in X direction.

d : The length of the offset link.

r : The radius of the wheel.

h : The distance between the robot and the hip coordinate systems.

θ_{s_i} : The angular distance between the robot and the hip coordinate systems.

For the C3P configuration the offset distance in equation (2.5) are

$$\begin{aligned} d_{s_{yi}} &= h\cos(\alpha_i) \\ d_{s_{xi}} &= h\sin(\alpha_i) \\ d_{y_i} &= d_{s_{yi}} + d\sin(\theta_{s_i}) \\ d_{x_i} &= d_{s_{xi}} + d\cos(\theta_{s_i}) \end{aligned} \tag{2.6}$$

The kinematic acceleration relation can be easily concluded from Appendix A by using the Jacobian proposed in [4] to obtain the robot acceleration solution.

$$\begin{gathered}\ddot{\mathbf{p}} = J_i\ \ddot{\mathbf{q}}_i\ +\ J_{r_i}\ \ddot{\mathbf{q}}_{r_i} \\ \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} -r\sin(\theta_{s_i}) & d_{y_i} & -d_{s_{yi}} \\ r\cos(\theta_{s_i}) & -d_{x_i} & d_{s_{xi}} \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_{x_i} \\ \ddot{\theta}_{c_i} \\ \ddot{\theta}_{s_i} \end{bmatrix} \\ + \begin{bmatrix} d_{x_i} & d_{s_{xi}} & d_{s_{xi}} \\ d_{y_i} & d_{s_{yi}} & d_{s_{yi}} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}^2_{c_i} \\ -2\ \dot{\theta}^2_{c_i}\dot{\theta}^2_{s_i} \\ \dot{\theta}^2_{s_i} \end{bmatrix}\end{gathered} \tag{2.7}$$

The first part of equation (2.7) is the acceleration component ($\ddot{\mathbf{q}}_i$), the centripetal velocities are ($\dot{\theta}^2_{c_i}$,$\dot{\theta}^2_{s_i}$), and ($-2\ \dot{\theta}^2_{c_i}\dot{\theta}^2_{s_i}$) are the Coriolis velocities where $i\ \in\ \{1,2,3\}$.

2.3.1 Inverse and Forward Kinematic Solutions

This section presents the kinematic solutions for the C3P. The methods are proposed by [4] and described in Appendix A. The main idea is to distinguish between the actuated and non actuated wheel's velocites. Furthermore, to distinguish between the sensed and nonsensed wheels velocities corresponding to the C3P configuration describtion in Section 2.3. The C3P has the wheels angular velocities $\dot{\theta}_{x_i}$ ($i\ \in\ \{1,2,3\}$) as the actuated robot elements. As a result the following actuated ($\dot{\mathbf{q}}_a$) and non actuated ($\dot{\mathbf{q}}_n$) wheel velocities vectors

$$\dot{\mathbf{q}}_a = \dot{\mathbf{q}}_x = \begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix}, \quad \dot{\mathbf{q}}_n = \begin{bmatrix} \dot{\mathbf{q}}_c \\ \dot{\mathbf{q}}_s \end{bmatrix} = \begin{bmatrix} \dot{\theta}_{c_1} \\ \dot{\theta}_{c_2} \\ \dot{\theta}_{c_3} \\ \dot{\theta}_{s_1} \\ \dot{\theta}_{s_2} \\ \dot{\theta}_{s_3} \end{bmatrix}, \tag{2.8}$$

correspondingly, the actuated (J_{a_i}) and non actuated (J_{n_i}) wheel Jacobians are

$$J_{a_i} = \begin{bmatrix} -r\sin(\theta_{s_i}) \\ r\cos(\theta_{s_i}) \\ 0 \end{bmatrix}, \quad J_{n_i} = \begin{bmatrix} h\cos(\alpha_i) + d\sin(\theta_{s_i}) & -h\cos(\alpha_i) \\ -h\sin(\alpha_i) + d\cos(\theta_{s_i}) & h\sin(\alpha_i) \\ 1 & -1 \end{bmatrix}. \quad (2.9)$$

After using the method in [4], which is descibed in Appendix A the actuated inverse solution for actuating the wheel angular velocities $\mathbf{q}_x$ will be described through the following equation

$$\dot{\mathbf{q}}_x = J_{in_x}\dot{\mathbf{p}}$$
$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta_{s_1}) & \cos(\theta_{s_1}) & h\cos(\alpha_1 - \theta_{s_1}) \\ -\sin(\theta_{s_2}) & \cos(\theta_{s_2}) & h\cos(\alpha_2 - \theta_{s_2}) \\ -\sin(\theta_{s_3}) & \cos(\theta_{s_3}) & h\cos(\alpha_3 - \theta_{s_3}) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}. \quad (2.10)$$

Furthermore, the inverse kinematic solution for actuating the steering angular velocities $\dot{\mathbf{q}}_s$ is

$$\dot{\mathbf{q}}_s = J_{in_s}\dot{\mathbf{p}}$$
$$\begin{bmatrix} \dot{\theta}_{s_1} \\ \dot{\theta}_{s_2} \\ \dot{\theta}_{s_3} \end{bmatrix} = \frac{-1}{d} \begin{bmatrix} \cos(\theta_{s_1}) & \sin(\theta_{s_1}) & -h\sin(\alpha_1 - \theta_{s_1}) + d \\ \cos(\theta_{s_2}) & \sin(\theta_{s_2}) & -h\sin(\alpha_2 - \theta_{s_2}) + d \\ \cos(\theta_{s_3}) & \sin(\theta_{s_3}) & -h\sin(\alpha_3 - \theta_{s_3}) + d \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (2.11)$$

and the inverse solution for $\dot{\mathbf{q}}_c$ is

$$\dot{\mathbf{q}}_c = J_{in_c}\dot{\mathbf{p}}$$
$$\begin{bmatrix} \dot{\theta}_{c_1} \\ \dot{\theta}_{c_2} \\ \dot{\theta}_{c_3} \end{bmatrix} = \frac{-1}{d} \begin{bmatrix} -\sin(\theta_{s_1}) & \cos(\theta_{s_1}) & -h\cos(\alpha_1 - \theta_{s_1}) \\ -\sin(\theta_{s_2}) & \cos(\theta_{s_2}) & -h\cos(\alpha_2 - \theta_{s_2}) \\ -\sin(\theta_{s_3}) & \cos(\theta_{s_3}) & -h\cos(\alpha_3 - \theta_{s_3}) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (2.12)$$

Equation (2.11) and (2.12) are used for further modeling equations.

The sensed velocities of wheel $i \in (1, 2, 3)$ are the wheel angular velocity $\dot{\theta}_{x_i}$ and the steering angular velocity $\dot{\theta}_{s_i}$, which gives the following sensed ($\dot{\mathbf{q}}_s$)and

nonsensed ($\dot{\mathbf{q}}_u$) velocity vectors

$$\dot{\mathbf{q}}_u = \dot{\mathbf{q}}_c, \quad \dot{\mathbf{q}}_s = \begin{bmatrix} \dot{\mathbf{q}}_x \\ \dot{\mathbf{q}}_s \end{bmatrix}, \tag{2.13}$$

and the sensed (J_{s_i}) and nonsensed (J_{u_i}) wheel Jacobians are

$$J_{s_i} = \begin{bmatrix} -r\sin(\theta_{s_i}) & -h\cos(\alpha_i) \\ r\cos(\theta_{s_i}) & h\sin(\alpha_i) \\ 0 & -1 \end{bmatrix}, \quad J_{u_i} = \begin{bmatrix} h\cos(\alpha_i) + d\sin(\theta_{s_i}) \\ -h\sin(\alpha_i) + d\cos(\theta_{s_i}) \\ 1 \end{bmatrix} \tag{2.14}$$

where, ($i \in \{1, 2, 3\}$). The sensed forward solution can be easily found from the solution of equation (A.8), where the sensed and nonsensed wheel velocities gives a robust sensing environment with possibility of slip detection. The forward kinematics is described as the following

$$\dot{\mathbf{p}} = J_{f_x}\dot{\mathbf{q}}_x + J_{f_s}\dot{\mathbf{q}}_s, \tag{2.15}$$

where J_{f_x} and J_{f_s} are the sensed forward solutions for the wheel angular and steering angular velocities.

$$J_{f_x} = \frac{1}{3}\begin{bmatrix} -r\sin(\theta_{s_1}) & -r\sin(\theta_{s_2}) & -r\sin(\theta_{s_3}) \\ r\cos(\theta_{s_1}) & r\cos(\theta_{s_2}) & r\cos(\theta_{s_3}) \\ 0 & 0 & 0 \end{bmatrix} \tag{2.16}$$

and

$$J_{f_s} = \frac{1}{3}\begin{bmatrix} -h\cos(\alpha_1) & -h\cos(\alpha_2) & -h\cos(\alpha_3) \\ h\sin(\alpha_1) & h\sin(\alpha_2) & h\sin(\alpha_3) \\ -1 & -1 & -1 \end{bmatrix} \tag{2.17}$$

The derivative of equation (2.15) yields the robot accelerations,

$$\ddot{\mathbf{p}} = J_{f_x}\ddot{\mathbf{q}}_x + J_{f_s}\ddot{\mathbf{q}}_s + g(\mathbf{q}_s, \dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s), \tag{2.18}$$

where

$$\ddot{\mathbf{p}} = \frac{d\dot{\mathbf{p}}}{dt}, \ \ddot{\mathbf{q}}_x = \frac{d\dot{\mathbf{q}}_x}{dt}, \ \ddot{\mathbf{q}}_s = \frac{d\dot{\mathbf{q}}_s}{dt}, \ \ddot{\mathbf{q}}_c = \frac{d\dot{\mathbf{q}}_c}{dt}. \tag{2.19}$$

2.4 Robot Dynamics Modeling

The dynamic model is considered as a complex problem to solve, which can be separated into two main problems: the inverse and forward dynamic problem. The inverse dynamic problem aims at determining the driven forces that produce specific motions, as well as the reactions which appear at each part of the multibody system's joints. The forward dynamic problem yields the motion of a multibody system over a given time interval, as a consequence of the applied forces and given initial conditions. The direct dynamic problem allows one to simulate and predict the system's actual behavior; motion is always the result of the forces that produce it.

In order to solve these two problems, the mobile robot is split in an open chain multibody system. The equation of motion for each part is obtained separately using Euler-Lagrangian method; then the platform constraints incorporate them into closed chain system with respect to the actuated variables.

The dynamic system can be classified as constrained and nonconstrained. Constraints imposed on a dynamic system may be holonomic, nonholonomic or both. This section shows whether the C3P wheeled mobile robot has holonomic constraints or not. There are several methods proposed in the literature mainly on the Frobenius Theorem [83]. Some reserchers, like those [12], used such a theorem to develop a method for determining any robotics system constraints. The method used is concrete and has been adopted in other literature [85] [86].

2.4.1 Nonholonomically Constrained System

We consider mechanical systems that are subject to the m velocity level equality type of nonholonomic constraints characterized by

$$B_n(\mathbf{q}_g)\dot{\mathbf{q}}_g = 0 \tag{2.20}$$

where $\mathbf{q}_g$ is the n-dimensional generalized coordinate, $B_n(\mathbf{q}_g)$ is an mxn dimensional matrix. Since the constraints are assumed to be nonholonomic, (2.20) then these constraints are independent. In another words, $B_n(\mathbf{q}_g)$ has rank m. It is noted that most nonholonomic constraints encountered in mechanical systems,

including rolling constraints, are in the form of (2.20).

Using the Lagrange multiplier rule, the equations of motion of nonholonomically constrained systems are governed by

$$M(\mathbf{q}_g)\ddot{\mathbf{q}}_g + G(\mathbf{q}_g, \dot{\mathbf{q}}_g) + C(\mathbf{q}_g) = \tau + B_n^T(\mathbf{q}_g)\lambda_n \tag{2.21}$$

where $M(\mathbf{q}_g)$ is the nxn dimensional positive definite inertia matrix, $G(\mathbf{q}_g, \dot{\mathbf{q}}_g)$ is the n-dimensional velocity-dependent force vector, $C(\mathbf{q}_g)$ is the the gravitational force vector, τ is the a_r-dimensional vector of actuator force/torque.

2.4.2 Holonomically Constrained System

We now assume that mechanical systems are subject to k holonomic constraints characterized by

$$B_h(\mathbf{q}_g) = \begin{bmatrix} b_{h_1}(\mathbf{q}_g) \\ b_{h_2}(\mathbf{q}_g) \\ . \\ . \\ b_{h_k}(\mathbf{q}_g) \end{bmatrix} = 0 \tag{2.22}$$

The equations of motion of the holonomically constrained system can also be obtained by using the Lagrange multiplier rule. They are given by

$$M(\mathbf{q}_g)\ddot{\mathbf{q}}_g + G(\mathbf{q}_g, \dot{\mathbf{q}}_g) + C(\mathbf{q}_g) = \tau + B_J^T(\mathbf{q}_g)\lambda_h \tag{2.23}$$

where λ_h is a k dimensional vector of Lagrange multipliers, and $B_J(\mathbf{q}_g)$ is the Jacobian of the holonomic constraints that is,

$$B_J(\mathbf{q}_g) = \frac{\partial B_h(\mathbf{q}_g)}{\partial \mathbf{q}_g} \tag{2.24}$$

We assumed that nonholonomic constraints are given by velocity-level equation (2.20) and holonomic constraints are described by position-level equation (2.22). The holonomic constraint is differentiated once and is represented at velocity level in the form of

$$\frac{dB_h(\mathbf{q}_g)}{dt} \quad \Rightarrow \quad B_J(\mathbf{q}_g)\dot{\mathbf{q}}_g = 0 \tag{2.25}$$

The velocity-level constraints in (2.25) are equivalent to the position constraint in (2.22), provided that the initial condition of the system $\mathbf{q}_{g_o} = \mathbf{q}_g(t_o)$ is a valid one, termed $B_h(\mathbf{q}_{g_o})$

In practical problems, both types of constraints may be described at velocity level. If both types of constraints are represented in the form of (2.20), the number of holonomic and nonholonomic constraints can be determined by using the Frobenius Theorem [84]. Briefly, let Δ_r be the distribution spanned by the null space of $B_n(\mathbf{q}_g)$, and Δ_r^* be the smallest involutive distribution containing Δ_r. It is clear that $n - m = \dim(\Delta_r) \leq \dim(\Delta_r^*)$. There are three possibilities.

1. If Δ_r itself is involutive, i.e., $\dim(\Delta_r) = \dim(\Delta_r^*)$, all m constraints are in fact holonomic.

2. If $\dim(\Delta_r^*) = n$, i.e., Δ_r^* spans the entire space, all m constraints are nonholonomic.

3. If $\dim(\Delta_r^*) = n - k$; k out of m constraints are holonomic and the remaining ones are nonholonomic.

If holonomic constraints are initially forced to be represented at velocity-level equations, then these constraints can be treated in the same way as nonholonomic constraints.

2.4.3 The C3P Platform Constrained System

This section will define whether the C3P has holonomic constraints. There are three variables describing the position and orientation of the platform ($x,\ y,\ \phi$) and three angles specifying the angular position of the driving wheels ($\theta_{x_1},\ \theta_{x_2},\ \theta_{x_3}$) in addition to three steering angles specifing the steering position of each caster wheel ($\theta_{s_1},\ \theta_{s_2},\ \theta_{s_3}$). Therefore the generalized coordinates are

$$\mathbf{q}_g = \begin{bmatrix} x & y & \phi & \theta_{x_1} & \theta_{x_2} & \theta_{x_3} & \theta_{s_1} & \theta_{s_2} & \theta_{s_3} \end{bmatrix}^T \tag{2.26}$$

where $n = 9$. Assuming the driving wheels roll (and do not slip), there are three constraints concluded from geometrical relation between each wheel varibales and the robot variables. The geometric relation of each wheel is presented by a contraint equation concluded from the caster wheel Jachobian (2.5). The constaints are

$$\begin{array}{c} -\dot{x} - \frac{1}{3}\sum_{i=1}^{3}(r\sin(\theta_{s_i})\dot{\theta}_{x_i}) - \frac{1}{3}\sum_{i=1}^{3}(h\cos(\alpha_i)\dot{\theta}_{s_i}) = 0 \\ -\dot{y} + \frac{1}{3}\sum_{i=1}^{3}(r\cos(\theta_{s_i})\dot{\theta}_{x_i}) + \frac{1}{3}\sum_{i=1}^{3}(h\sin(\alpha_i)\dot{\theta}_{s_i}) = 0 \\ -\dot{\phi} - \frac{1}{3}\sum_{i=1}^{3}(\dot{\theta}_{s_i}) = 0 \end{array} \tag{2.27}$$

where $k = 3$. The three constraints ($k = 3$) can be written in the form of:

$$B_J(\mathbf{q}_g)\dot{\mathbf{q}}_g = 0 \tag{2.28}$$

where
$B_J(\mathbf{q}_g)\dot{\mathbf{q}}_g = \frac{1}{3}$

$$\begin{bmatrix} -3 & 0 & 0 & -rS(\theta_{s_1}) & -rS(\theta_{s_2}) & -rS(\theta_{s_3}) & -hC(\alpha_1) & -hC(\alpha_2) & -hC(\alpha_3) \\ 0 & -3 & 0 & rC(\theta_{s_1}) & rC(\theta_{s_2}) & rC(\theta_{s_3}) & hS(\alpha_1) & hS(\alpha_2) & hS(\alpha_3) \\ 0 & 0 & -3 & 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \\ \dot{\theta}_{s_1} \\ \dot{\theta}_{s_2} \\ \dot{\theta}_{s_3} \end{bmatrix} \tag{2.29}$$

The matrix $B_J(\mathbf{q}_g) \in R^{kXn}(R^{3X9})$. The matrix span $S(\mathbf{q}_g) \in R^{nX(n-k)}(R^{9X6})$ spans the null space of $B_J(\mathbf{q}_g)$ and a full-rank matrix formed by a set of smooth and linearly independent vector fields, $\upsilon \in R^{n-k}(R^6)$, where $\dot{\mathbf{q}}_g = S(\mathbf{q}_g)\upsilon$ and $\upsilon = [\dot{\theta}_{x_1}\ \dot{\theta}_{x_2}\ \dot{\theta}_{x_3}\ \dot{\theta}_{s_1}\ \dot{\theta}_{s_2}\ \dot{\theta}_{s_3}]^T$. The matrix span can be concluded directly from the constraint equations (2.27)

$S(\mathbf{q}_g) = \frac{1}{3}$

$$\begin{bmatrix} -rS(\theta_{s_1}) & -rS(\theta_{s_2}) & -rS(\theta_{s_3}) & -hC(\alpha_1) & -hC(\alpha_2) & -hC(\alpha_3) \\ rC(\theta_{s_1}) & rC(\theta_{s_2}) & rC(\theta_{s_3}) & hS(\alpha_1) & hS(\alpha_2) & hS(\alpha_3) \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 3 & . & . & . & . & . \\ . & 3 & . & . & . & . \\ . & . & 3 & . & 0 & . \\ . & . & . & 3 & . & . \\ . & 0 & . & . & 3 & . \\ . & . & . & . & . & 3 \end{bmatrix} \tag{2.30}$$

$S(\mathbf{q}_g) = \begin{bmatrix} \mathbf{s}(\mathbf{q}_g)_1 & \mathbf{s}(\mathbf{q}_g)_2 & \mathbf{s}(\mathbf{q}_g)_3 & \mathbf{s}(\mathbf{q}_g)_4 & \mathbf{s}(\mathbf{q}_g)_5 & \mathbf{s}(\mathbf{q}_g)_6 \end{bmatrix}$

Let Δ_r be the distribution spanned by the vectors $\mathbf{s}(\mathbf{q}_g)_1$, $\mathbf{s}(\mathbf{q}_g)_2$, $\mathbf{s}(\mathbf{q}_g)_3$, $\mathbf{s}(\mathbf{q}_g)_4$,$\mathbf{s}(\mathbf{q}_g)_5$ and $\mathbf{s}(\mathbf{q}_g)_6$

$$\Delta_r = span\left\{\mathbf{s}(\mathbf{q}_g)_1, \mathbf{s}(\mathbf{q}_g)_2, \mathbf{s}(\mathbf{q}_g)_3, \mathbf{q}_g)_4, \mathbf{s}(\mathbf{q}_g)_5, \mathbf{s}(\mathbf{q}_g)_6\right\} \tag{2.31}$$

Taking the Lie Bracket for any pair of the vectors from the distribution Δ_r, if each resultant vector field is still contained in Δ_r, then the distribution is involutive, where the Lie Bracket rule is

$$[\mathbf{s}(\mathbf{q}_g)_i, \mathbf{s}(\mathbf{q}_g)_j] = \frac{\mathbf{s}(\partial \mathbf{q}_g)_i}{\partial \mathbf{q}_g}\mathbf{s}(\mathbf{q}_g)_j - \frac{\mathbf{s}(\partial \mathbf{q}_g)_j}{\partial \mathbf{q}_g}\mathbf{s}(\mathbf{q}_g)_i \tag{2.32}$$

The result of the Lie Bracket rule is

$$[\mathbf{s}(\mathbf{q}_g)_i, \mathbf{s}(\mathbf{q}_g)_j] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{2.33}$$

where $\{i, j\} \in (1-6)$ and $i \neq j$. It is noticable that all the resultant vectors are lineary dependent on Δ_r, since the zero vector belongs to any vector distribution. Where $[\mathbf{s}(\mathbf{q}_g)_i, \mathbf{s}(\mathbf{q}_g)_j] \in \Delta_r$ and $i, j \in (1-6), i \neq j$. As a result, the distribution Δ_r is involutive and dim (Δ_r^*)=dim(Δ_r)=$n-k$=6. Therefore, all the constraints are holonomic, which is the first possibility of the criterion mentioned in Section

(2.4.2). As a conclusion, the C3P has holonomic constraints and the Lagrangian quation will be

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_g}\right) - \frac{\partial L}{\partial \mathbf{q}_g} = \tau + B_J^T(\mathbf{q}_g)\lambda_h \tag{2.34}$$

The constriant equations describe the relation between platform variables $\mathbf{p} = \mathbf{p}(x, y, z)$ and the wheel's angular velocities $\mathbf{q}_x = \mathbf{q}_x(\theta_{x_1}, \theta_{x_2}, \theta_{x_3})$. If the generalized coordinates vector $\mathbf{q}_g = \mathbf{q}_x$, then the Lagrangian equation will be

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_x}\right) - \frac{\partial L}{\partial \mathbf{q}_x} = \tau \tag{2.35}$$

2.4.4 Euler-Lagrange Method

This section presents the forced Euler-Lagrange [78] equations of motion which are used in deriving the C3P motion equation. Simple mechanical systems described by the Lagrangian form is the difference between a kinetic energy (K) and potential energy (P)

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K - P = \frac{1}{2}\,\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}} - V(\mathbf{q}) \tag{2.36}$$

where $\mathbf{q}$ is the particle position vector which belongs to the generalized robot coordinate, $M(\mathbf{q})$ is the inertia matrix which is a positive definite symmetric matrix, and $V(\mathbf{q})$ is the potential energy of the system. Since many wheel mobile robots are assumed to move on a horizontal planar surface, P is zero and the Lagrangian function of the WMR is the sum of the robot parts and joints K.

$$L = \sum_{i=1}^{N} K_i, \tag{2.37}$$

where N is the number of parts in the robot. The Lagrangian function is used to obtain the Lagrangian dynamic formulation which is described as

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial \mathbf{q}} \tag{2.38}$$

The generalized coordinates $\mathbf{q}$ vector contains the actuated displacements variables and $\dot{\mathbf{q}}$ vecotr contains the actuated velocities, while τ contains the external torque/force vector. The overall dynamics of the robot can be formulated as a

system of ordinary differential equations whose solutions are required to satisfy the WMR constraints as the following

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + G\left(\mathbf{q}, \dot{\mathbf{q}}\right) + C(\mathbf{q}) \tag{2.39}$$

where, $M(\mathbf{q})$ is the inertia matrix, $G\left(\mathbf{q}, \dot{\mathbf{q}}\right)$ is the centripetal and Coriolis velocity terms, and $C(\mathbf{q})$ is the vector containing gravity terms.

2.4.5 Kinetic Energy Equations

The kinetic energy of the rigid body depends on its mass, inertia, linear and angular velocities as described in the following equation

$$K = \frac{1}{2}\, m\ \boldsymbol{V}^T\ \boldsymbol{V} + \frac{1}{2}\, \Omega^T\ I\ \Omega \tag{2.40}$$

where,

m, I : mass and inertia of the rigid body.

$\boldsymbol{V}$, Ω : linear and angular velocity at the center of gravity of the rigid body.

The Kinetic Energy of a rigid body is equal to the kinetic energy of a point mass m located at the center mass. The linear and angular velocity vectors of each part are obtained from the forward kinematic relation depending on their center of gravity. The platform velocity constraints are used to incorporate the open chain dynamics into a closed chain dynamics to deliver the Lagrangian equation.

2.5 Dynamic Modeling

The Lagrangian formulation is used to obtain the C3P forward dynamics considering the robot as a multibody closed-chain system with mobility constraints. The robot structure consists of seven main parts: three wheels, three links and one platform (Fig. 2.3). The motor and steering mechanisms along with the wheel slippage and the friction (contact point and bearing of the steering axes) are ignored in the proposed dynamic model.

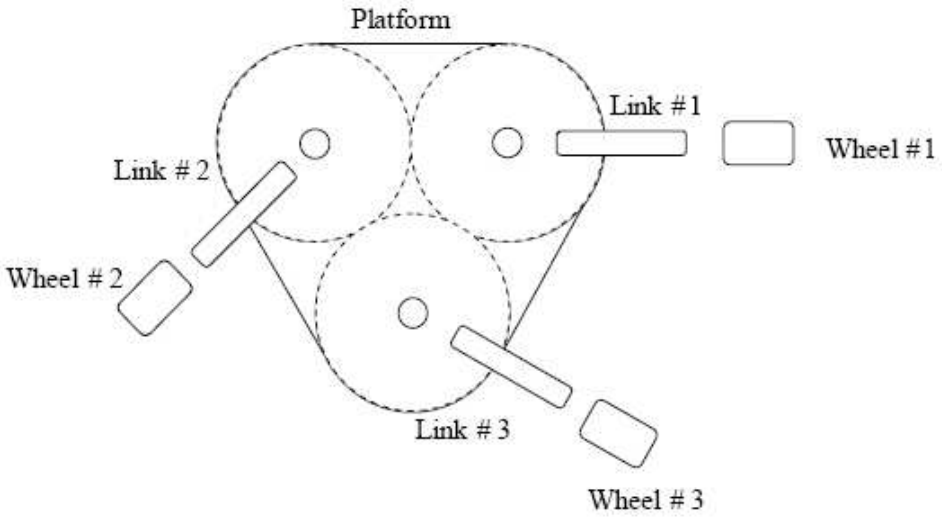

Figure 2.3. The C3P parts structure

In this work, wheels and links mass and inertia have negligible value in comparison to the mass and inertia of the C3P platform. Therefore, their kinetic energy will not be included in the Largangian equation.

The C3P dynamic model consists of two main equations of motion: The Wheels Torque Dynamics (WTD) and the Dynamic Steering Estimator (DSE). The wheel angular velocities $\dot{\mathbf{q}}_x$ are calculated using the (WTD) equation of motion with respect to the actuated wheels torques τ_x. The steering angles are not controlled and their behaviour is affected by wheel angular velocities $\dot{\mathbf{q}}_x$ values. Therefore the steering angles and the steering angular wheel velocities are recursively calculated by the (DSE) corresponding to the angular wheels velocities and accelerations.

2.5.1 The Wheels Torque Dynamics (WTD)

The Wheels Torque Dynamic model is the relation between the applied torques on the wheels τ_x and the output wheel velocities $\dot{\mathbf{q}}_x$. The dynamics equation depends mainly on the platform mobility constraints, which appears in the robot velocities equation. The translational and angular velocities are obtained from the forward kinematics equation (2.16), with respect to the wheel angular velocities ($\dot{\mathbf{q}}_x$) as the

actuated variables. The velocities are obtained symbolically by the aids of Maple and are described as the following

$$\boldsymbol{V}_{p(WTD)} = \left[\sum_{i=1}^{3} -r\ \sin(\theta_{s_i})\ \dot{\theta}_{x_i},\ \ \sum_{i=1}^{3} r\ \cos(\theta_{s_i})\ \dot{\theta}_{x_i}\right]^T /3 \tag{2.41}$$

and

$$\Omega_{p(WTD)} = \left[0\ \dot{\theta}_{x_i}\right] \tag{2.42}$$

To be substituted in the kinetic engery equation (2.40)and the platform kinetic energy will be

$$K_{p(WTD)} = \frac{1}{2}\ m_p\ \boldsymbol{V}^T_{p(WTD)}\ \boldsymbol{V}_{p(WTD)} + \frac{1}{2}\ \Omega^T_{p(WTD)}\ I_p\ \Omega_{p(WTD)} \tag{2.43}$$

where m_p is the wheel mass and I_p is the wheel inertia located at the platform center of gravity, which is assumed to be located at the robot geometric coordinates. Then the Lagrangian function will be $L = K_{p(WTD)} = f(\mathbf{q}_s, \dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s)$. The subjected torques are applied to the wheel angular velocities $\mathbf{q}_x$. Therefore the generalized velocities will be $\mathbf{q}_g = \mathbf{q}_x$ and the Lagrangian formulation is

$$\tau_x = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_x}\right) - \frac{\partial L}{\partial \mathbf{q}_x} \tag{2.44}$$

and the forward dynamic equation can be represented in the following equation

$$\tau_x = M(q_s)\ddot{\mathbf{q}}_x, \tag{2.45}$$

where $\tau_x = \left[\tau_{x_1}\ \tau_{x_2}\ \tau_{x_3}\right]^T$ vector is a function of angular wheel accelerations $\ddot{q}_x$ and the steering angles q_s.

$$\tau_x = M\ \ddot{\mathbf{q}}_x = m_p r^2 \begin{bmatrix} 1 & \cos(\theta_{s_1} - \theta_{s_2}) & \cos(\theta_{s_1} - \theta_{s_3}) \\ \cos(\theta_{s_1} - \theta_{s_2}) & 1 & \cos(\theta_{s_2} - \theta_{s_3}) \\ \cos(\theta_{s_1} - \theta_{s_3}) & \cos(\theta_{s_2} - \theta_{s_3}) & 1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_{x_1} \\ \ddot{\theta}_{x_2} \\ \ddot{\theta}_{x_3} \end{bmatrix} \tag{2.46}$$

2.5.2 The Dynamic Steering Estimator (DSE)

The dynamic steering estimator (DSE) predicts the values of the steering angular accelerations, velocities, and angles. Therefore, the steering angluar velocities $\dot{\mathbf{q}}_s$ will be considered as the actuated velocites.The translational and angular robot velocities obtained from the forward kinematic solution (2.17) is represented as the following

$$\boldsymbol{V}_{p(DSE)} = \left[-\sum_{i=1}^{3} (r\sin(\theta_{s_i})\dot{\theta}_{x_i} + h\cos(\alpha_i)\dot{\theta}_{s_i}), \quad \sum_{i=1}^{3} (r\cos(\theta_{s_i})\dot{\theta}_{x_i} + h\sin(\alpha_i)\dot{\theta}_{s_i}) \right]^T /3 \tag{2.47}$$

$$\Omega_{p(DSE)} = \left[-\sum_{i=1}^{3} \dot{\theta}_{s_i} \right] /3 \tag{2.48}$$

Equation (2.47) and (2.48) are used along with the kinetic engery equation (2.40) to abtain the following platform kinetic energy

$$K_{p(DSE)} = \frac{1}{2}\, m_p\ \boldsymbol{V}^T_{p(DSE)}\ \boldsymbol{V}_{p(DSE)} + \frac{1}{2}\, \Omega^T_{p(DSE)}\ I_p\ \Omega_{p(DSE)} \tag{2.49}$$

Corresponding to the Lagrangian function $L = K_{p(DSE)} = f(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_x)$ and generalized velocities are $\mathbf{q}_g = \mathbf{q}_s$, the Lagrangian formulation will be

$$\tau_s = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_s}\right) - \frac{\partial L}{\partial \mathbf{q}_s} \tag{2.50}$$

which results the torque equation

$$\tau_s = M_{sx}\ddot{\mathbf{q}}_{sx} + G_{sx}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{2.51}$$

The inertia matrix M_{sx} can be decomposed into two matrices; M_x contains the mass and inertia parameters related to wheels angular accelerations $\ddot{\mathbf{q}}_x$, the matrix M_s contains the parameters related to the steering angular accelerations $\ddot{\mathbf{q}}_s$

$$\tau_s = \begin{bmatrix} M_x & M_s \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{bmatrix} + G_{sx}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{2.52}$$

where,

$$\tau_{s_{(3x1)}} = M_{x_{(3x3)}}\ \ddot{\mathbf{q}}_{x_{(3x1)}} + M_{s_{(3x3)}}\ \ddot{\mathbf{q}}_{s_{(3x1)}} + G_{sx_{(3x1)}}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{2.53}$$

The matrixes M_x, M_s and the vector G_{sx} are described in Appendix (B). Using the virtual work principle, the Jacobian of actuating the steering angluar velocity, described by equation(2.11), will project the platform main forces/torques vector τ_{pl} to the steering angular torques vector τ_s

$$\tau_s = J_{in_s}\tau_{p_l}, \tag{2.54}$$

where

$$\tau_{pl} = M_{p_l}\ddot{\mathbf{p}}, \tag{2.55}$$

and

$$J_{in_s} = \begin{bmatrix} -\cos(\theta_1) & -\sin(\theta_1) & h\sin(\alpha_1-\theta_1)-d \\ -\cos(\theta_2) & -\sin(\theta_2) & h\sin(\alpha_2-\theta_2)-d \\ -\cos(\theta_3) & -\sin(\theta_3) & h\sin(\alpha_3-\theta_3)-d \end{bmatrix}/d \tag{2.56}$$

$$\tau_{p_l} = \begin{bmatrix} F_x \\ F_y \\ \tau_\phi \end{bmatrix},\ M_{p_l} = \begin{bmatrix} m_p & 0 & 0 \\ 0 & m_p & 0 \\ 0 & 0 & I_p \end{bmatrix}. \tag{2.57}$$

The sensing forward acceleration solution for the C3P is

$$\ddot{\mathbf{p}} = J_{f_x}\ddot{\mathbf{q}}_x + J_{f_s}\ddot{\mathbf{q}}_s + g(\mathbf{q}_s, \dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s) \tag{2.58}$$

by substituting (2.58) and (2.55) in (2.54)

$$\tau_s = J_{in_s}M_{p_l}J_{f_s}\ddot{\mathbf{q}}_s + J_{in_s}M_{p_l}J_{f_x}\ddot{\mathbf{q}}_x + J_{in_s}M_{p_l}\ g(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_x). \tag{2.59}$$

From (2.59) and (2.52)

$$\ddot{\mathbf{q}}_s = M_{ss}^{-1}M_{xx}\ddot{\mathbf{q}}_x + M_{ss}^{-1}G_{ssx}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_x) \tag{2.60}$$

with

$$M_{ss} = J_{in_s}M_{p_l}J_{f_s} - M_s \tag{2.61}$$

$$M_{xx} = M_x - J_{in_s}M_{p_l}J_{f_x} \tag{2.62}$$

$$G_{ssx}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_x) = G_{sx} - J_{in_s} M_{pl}\ g(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_x). \tag{2.63}$$

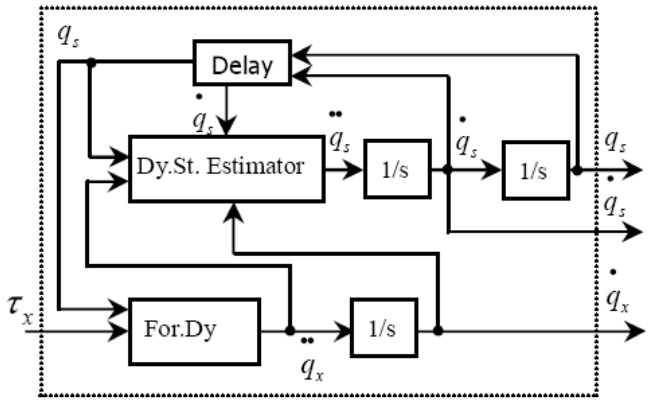

Figure 2.4. C3P Dynamic model

After deriving the dynamic equations of the C3P, it is revealed that the dynamic steering estimator calculates the steering angle and the steering angular velocity, which are affected by the wheel angular acceleration resulted from the forward dynamic equation. Such a structure is represented in Figure 2.4.

The C3P dynamic model has the actuated wheel torques τ_x as an input while the outputs are the sensed wheel velocities $\dot{\mathbf{q}}_x$, the steering angular velocities $\dot{\mathbf{q}}_s$, and the steering angles $\mathbf{q}_s$. Since the steering angular velocities are actuated by the angular wheel velocities, the angular wheel velocities $\dot{\mathbf{q}}_x$ and accelerations $\ddot{\mathbf{q}}_x$ are the main inputs of the steering dynamic estimator. The steering angles $\mathbf{q}_s$ and steering angular velocities $\dot{\mathbf{q}}_s$ are delayed by unity time interval because the steering dynamic model is calculated recursively corresponding to (2.60).

Chapter 3

Kinematics Based Motion Control

The C3P modular configuration for the actuators can be easily built in the laboratory. On the other hand, a problem is found during the kinematics modeling. This problem is identified as kinematics singularities, which is formulated in Section (3.1). A modified inverse kinematics solution is proposed in Section (3.2) under the name of Wheel Coupling Equation (WCE) to solve the singularity problem. The solution depends on using the coupling action between each two wheel angular velocities to virtually actuate the steering angular velocity of the third wheel. Simulation examples are used in section (3.2.1) to show the feasibility of the approach. A regulator for the WCE is developed in Section (3.3). A velocity control structure is proposed in Section (3.4) and the simulation results are used to illustrate its performance.

3.1 C3P Singularities

The C3P has the wheel angular velocities $\dot{\theta}_{x_i}$ ($i \in \{1, 2, 3\}$) as the actuated robot wheels velocites. Corresponding to the actuated inverse solution obtained in Chapter (2), it is pretty noticable that the Jachobian matix J_{in_x} can be singular for some steering angles configurations.

$$\dot{\mathbf{q}}_x = J_{in_x}\dot{\mathbf{p}}$$

$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta_{s_1}) & \cos(\theta_{s_1}) & h\cos(\alpha_1 - \theta_{s_1}) \\ -\sin(\theta_{s_2}) & \cos(\theta_{s_2}) & h\cos(\alpha_2 - \theta_{s_2}) \\ -\sin(\theta_{s_3}) & \cos(\theta_{s_3}) & h\cos(\alpha_3 - \theta_{s_3}) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}. \tag{3.1}$$

The singularity effect appears only when the steering angles yield the same value. For example, when the steering angles are equal to -90^o, the robot is not actuated in the Y direction and the velocity $\dot{y}$ is zero (Fig. 3.1a), and when the angles are equal to 0^o the velocity $\dot{x}$ is zero (Fig. 3.1c). The steering configuration in Figure (3.1b) gives a singular determinant for the matrix J_{in_x} with -45^o steering angles although all the robot DOFs are actuated except the direction of (-X,Y). For example, the robot cannot move in the direction of $\dot{\mathbf{p}} = [-1(m/s)\ 1(m/s)\ 0(r/min)]^T$ is not actuated. This concludes the following: if all steering angles yield the same value, then the robot cannot be actuated in the direction perpendicular to the wheel angular velocity vector. Figure (3.1d) represents a non-singular steering wheels configuration condition.

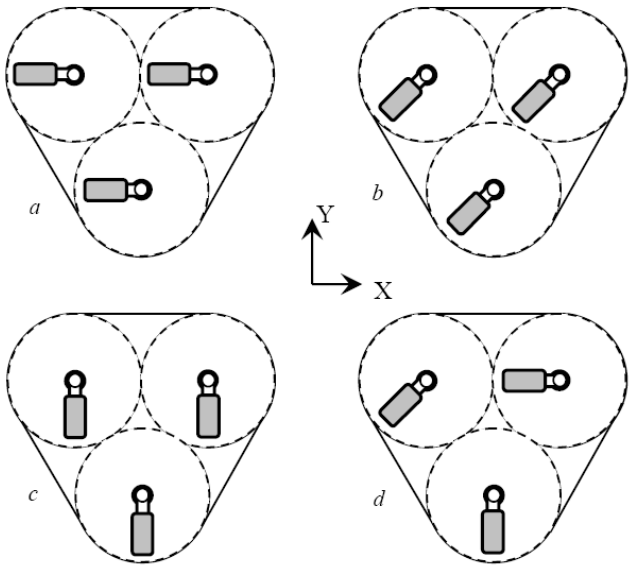

Figure 3.1. Different steering configurations

3.2 Coupling Approach

It is mentioned in [4] that if at least one variable of the wheel variables is not actuated, then the required mobility is not totally guaranteed. In the C3P, the steering angular velocity $\dot{\theta}_s$ of each castor wheel is not actuated, and as a result, singularitites are found in the in the Jacobian J_{in_x}. Therefore, the steering angular velocities $\dot{\mathbf{q}}_s$ should be virtually actuated by the effect of the wheels angular velocities $\dot{\mathbf{q}}_x$ one way or another.

In order to develop a singularity-free solution, the inverse kinematics solution for actuating the steering angular velocities (3.2) will be used within the actuated inverse solution (3.1).

$$\dot{\mathbf{q}}_s = J_{in_s}\dot{\mathbf{p}}$$

$$\begin{bmatrix} \dot{\theta}_{s_1} \\ \dot{\theta}_{s_2} \\ \dot{\theta}_{s_3} \end{bmatrix} = \frac{-1}{d} \begin{bmatrix} \cos(\theta_{s_1}) & \sin(\theta_{s_1}) & -h\sin(\alpha_1 - \theta_{s_1}) + d \\ \cos(\theta_{s_2}) & \sin(\theta_{s_2}) & -h\sin(\alpha_2 - \theta_{s_2}) + d \\ \cos(\theta_{s_3}) & \sin(\theta_{s_3}) & -h\sin(\alpha_3 - \theta_{s_3}) + d \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} . \tag{3.2}$$

As basic a idea, the coupling between the angular velocities of any two wheels virtually actuates the steering angular velocity of the third wheel (Fig.3.2). For example, the coupling between $\dot{\theta}_{x_1}$ and $\dot{\theta}_{x_2}$ actuates $\dot{\theta}_{s_3}$

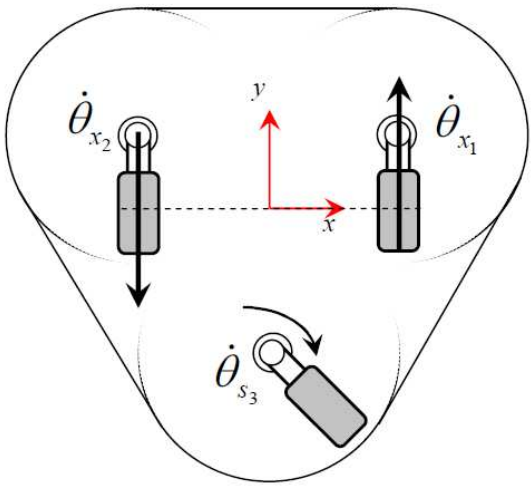

Figure 3.2. Coupling between $\dot{\theta}_{x_1}$ and $\dot{\theta}_{x_2}$

The Jacobian solution J_{in_x} (3.2) and J_{in_s} (3.1) are used to deliver a non-singular inverse kinematics Jacobian to virtually actuate each steering angle. This can be achieved by the following steps:

- Step 1: from equation (2.5) (caster wheel Jacobian) a relation between $\dot{\mathbf{p}}$ and $\dot{\theta}_{s_i}$ will be obtained as a link between equation (3.1)and equation (3.2).
- Step 2: decompose the wheels angular velocities actuation solution for wheels 1 and 2 ($\dot{\theta}_{x_1}$ and $\dot{\theta}_{x_2}$) from equation (3.1).
- Step 3: decompose the steering angular velocity solution for actuating wheel number 3 ($\dot{\theta}_{s_3}$) from equation (3.2).
- Step 4: combine steps (1), (2) and (3) to obtain a solution for actuating the steering angular velocity.
- Step 5: applying previous steps for all the wheels.
- Step 6: adding the obtained solution to (3.1).

Step 1, the actuated Jacobian for the i^{th} caster wheel is

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -d_{s_{yi}} \\ d_{s_{xi}} \\ -1 \end{bmatrix} \dot{\theta}_{s_i}, \tag{3.3}$$

under two main assumptions: 1) $\dot{\theta}_{x_i}$ has zero relative velocity to the robot velocity and 2) $\dot{\theta}_{c_i}$ has zero initial value because it is fixed on the floor for the first instant.

Step 2, for actuation $\dot{\theta}_{s_3}$ the inverse actuation solutions for $\dot{\theta}_{x_1}$ and $\dot{\theta}_{x_2}$ are used, which is decomposed from (3.1)

$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta_{s_1}) & \cos(\theta_{s_1}) & h\cos(\alpha_1 - \theta_{s_1}) \\ -\sin(\theta_{s_2}) & \cos(\theta_{s_2}) & h\cos(\alpha_2 - \theta_{s_2}) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}. \tag{3.4}$$

Step 3, the steering angular velocity value $\dot{\theta}_{s_3}$ is the function of θ_{s_3} and the reference robot velocities $\dot{\mathbf{p}}$. Such a relation is found in the inverse solution (3.2)

and $\dot{\theta}_{s_3}$ is concluded as shown below

$$\dot{\theta}_{s_3} = \frac{-1}{d} \begin{bmatrix} \cos(\theta_{s_3}) & \sin(\theta_{s_3}) & -h\sin(\alpha_3 + \theta_{s_3}) - d \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{3.5}$$

Step 4, from equations (3.3), (3.4) and (3.5), the following solution is obtained

$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \end{bmatrix} = J_{A_3} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{3.6}$$

where, $J_{A_3} \in R^{2X3}$

$$\begin{aligned} J_{A_3}(1,1) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_1} - \phi_3) - \cos(\theta_{s_1} - \phi_1)) * \cos(\theta_{s_3}) \\ J_{A_3}(2,1) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_2} - \phi_3) - \cos(\theta_{s_2} - \phi_2)) * \cos(\theta_{s_3}) \end{aligned} \tag{3.7}$$

$$\begin{aligned} J_{A_3}(1,2) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_1} - \phi_3) - \cos(\theta_{s_1} - \phi_1)) * \sin(\theta_{s_3}) \\ J_{A_3}(2,2) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_2} - \phi_3) - \cos(\theta_{s_2} - \phi_2)) * \sin(\theta_{s_3}) \end{aligned} \tag{3.8}$$

$$\begin{aligned} J_{A_3}(1,3) &= \tfrac{h}{r\ d}(\cos(\theta_{s_1} - \phi_3) - \cos(\theta_{s_1} - \phi_1)) * \sin(\theta_{s_3} + \phi_3) \\ J_{A_3}(2,3) &= \tfrac{h}{r\ d}(\cos(\theta_{s_2} - \phi_3) - \cos(\theta_{s_2} - \phi_2)) * \sin(\theta_{s_3} + \phi_3) \end{aligned} \tag{3.9}$$

Step 5, the same procedures are used for virtually actuating steering angular velocity $\dot{\theta}_{s_2}$ to obtain

$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_3} \end{bmatrix} = J_{A_2} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{3.10}$$

where, $J_{A_2} \in R^{2X3}$

$$\begin{aligned} J_{A_2}(1,1) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_1} - \phi_2) - \cos(\theta_{s_1} - \phi_1)) * \cos(\theta_{s_2}) \\ J_{A_2}(2,1) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_3} - \phi_2) - \cos(\theta_{s_3} - \phi_2)) * \cos(\theta_{s_2}) \end{aligned} \tag{3.11}$$

$$\begin{aligned} J_{A_2}(1,2) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_1} - \phi_2) - \cos(\theta_{s_1} - \phi_1)) * \sin(\theta_{s_2}) \\ J_{A_2}(2,2) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_3} - \phi_2) - \cos(\theta_{s_3} - \phi_3)) * \sin(\theta_{s_2}) \end{aligned} \tag{3.12}$$

$$\begin{aligned} J_{A_2}(1,3) &= \tfrac{h}{r\ d}(\cos(\theta_{s_1}-\phi_2)-\cos(\theta_{s_1}-\phi_1))*\sin(\theta_{s_2}+\phi_2) \\ J_{A_2}(2,3) &= \tfrac{h}{r\ d}(\cos(\theta_{s_3}-\phi_2)-\cos(\theta_{s_3}-\phi_3))*\sin(\theta_{s_2}+\phi_2) \end{aligned} \tag{3.13}$$

and also used for the steering angular velocity $\dot{\theta}_{s_1}$ to deliver

$$\begin{bmatrix} \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix} = J_{A_1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{3.14}$$

where, $J_{A_1} \in R^{2X3}$

$$\begin{aligned} J_{A_1}(1,1) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_2}-\phi_1)-\cos(\theta_{s_2}-\phi_2))*\cos(\theta_{s_1}) \\ J_{A_1}(2,1) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_3}-\phi_1)-\cos(\theta_{s_3}-\phi_2))*\cos(\theta_{s_1}) \end{aligned} \tag{3.15}$$

$$\begin{aligned} J_{A_1}(1,2) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_2}-\phi_1)-\cos(\theta_{s_2}-\phi_2))*\sin(\theta_{s_1}) \\ J_{A_1}(2,2) &= \tfrac{-h}{r\ d}(\cos(\theta_{s_3}-\phi_1)-\cos(\theta_{s_3}-\phi_3))*\sin(\theta_{s_1}) \end{aligned} \tag{3.16}$$

$$\begin{aligned} J_{A_1}(1,3) &= \tfrac{h}{r\ d}(\cos(\theta_{s_2}-\phi_1)-\cos(\theta_{s_2}-\phi_2))*\sin(\theta_{s_1}+\phi_1) \\ J_{A_1}(2,3) &= \tfrac{h}{r\ d}(\cos(\theta_{s_3}-\phi_1)-\cos(\theta_{s_3}-\phi_3))*\sin(\theta_{s_1}+\phi_1) \end{aligned} \tag{3.17}$$

Step 6 combines the solution (3.7), (3.11) and (3.15) with the J_{in_x} Jacobian found in equation (3.1) as the following

$$J_{S_1} = \begin{bmatrix} J_{in_x}(1,1) & J_{in_x}(1,2) & J_{in_x}(1,3) \\ J_{A_1}(1,1)\ m & J_{A_1}(1,2)\ m & J_{A_1}(1,3)\ m \\ J_{A_1}(2,1)\ m & J_{A_1}(2,2)\ m & J_{A_1}(2,3)\ m \end{bmatrix} \tag{3.18}$$

$$J_{S_2} = \begin{bmatrix} J_{A_2}(1,1)\ m & J_{A_2}(1,2)\ m & J_{A_2}(1,3)\ m \\ J_{in_x}(2,1) & J_{in_x}(2,2) & J_{in_x}(2,3) \\ J_{A_2}(2,1)\ m & J_{A_2}(2,2)\ m & J_{A_2}(2,3)\ m \end{bmatrix} \tag{3.19}$$

$$J_{S_3} = \begin{bmatrix} J_{A_3}(1,1)\ m & J_{A_3}(1,2)\ m & J_{A_3}(1,3)\ m \\ J_{A_3}(2,1)\ m & J_{A_3}(2,2)\ m & J_{A_3}(2,3)\ m \\ J_{in_x}(3,1) & J_{in_x}(3,2) & J_{in_x}(3,3) \end{bmatrix}, \tag{3.20}$$

Where the variable 'm' is a weighting value to regulate the wheel coupling effect, the Jacobian matrices J_{S_1}, J_{S_2} and J_{S_3} virtually actuate the wheel steering angular velocities $\dot{\theta}_{s_1}$, $\dot{\theta}_{s_2}$ and $\dot{\theta}_{s_3}$ respectively. The three Jacobian matrices are added to

each other in the following solution

$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix} = J_{WCE} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = [J_{S_1} + J_{S_2} + J_{S_3}] \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}, \tag{3.21}$$

Each wheel virtually actuates the steering angular velocity of the other two wheels at the same time, thus the selected value of m is 0.5 or $\frac{1}{2}$ corresponding to the average value resulted for actuating two steering angular velocities simultaneously. For the WCE (Wheel Coupling Equation) (3.21) the wheel coupling produces distortions of the exact solution in the proximity of singularities. If the wheel configuration is far from singular condition, the solution in equation (3.21) reaches asymptotically the inverse kinematics J_{in_x} solution in equation (3.2).

3.2.1 Simulation Examples

For the simulation examples in this section, the open loop control structure shown in Figure (3.3) is concerned. The axes level loop controls the wheel angular velocities $\dot{\mathbf{q}}_x$ with respect to its reference value $\dot{\mathbf{q}}_{xr}$. The Wheel Coupling Equation represented by the block 'WCE' is the non-singular kinematics explained in Section (3.2).

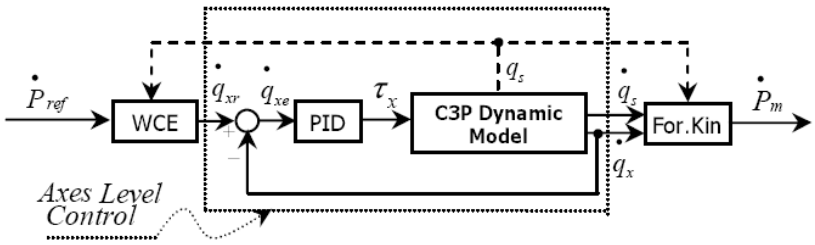

Figure 3.3. Open loop structure using C3P dynamic model

The parameters of the robot, used in this chapter, are described in Table. 3.1. These parameters are used for all the examples presented in this chapter.

The first example represents a singular initial condition, which means the following: a) the steering angles are parallel to each other, b) the desired direction of motion is perperndicular on the wheels axes as shown in Figure(3.4). The initial

Table 3.1. The C3P parameters

C3P Parameters	Value	Units
h	0.343	m
d	0.04	m
r	0.04	m
M_p (P_l mass)	30	Kg
I_p (P_l inertia)	3.51	$Kg\ m^2$

value of the steering angles is 45^o (Fig .(3.4-a)) and the desired direction of motion is $\dot{\mathbf{p}} = [0.12(m/s)\ 0.12(m/s)\ 0(r/min)]^T$. For such input, the steering angles should yield the value of -45^o as shown in Figure(3.4-b).

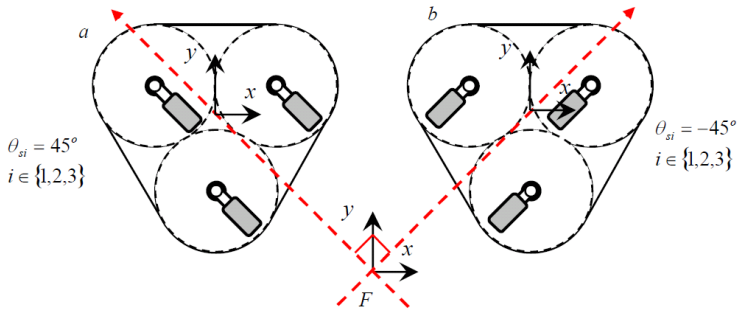

Figure 3.4. C3P wheel configuration considering the results in Figure (3.5)

Figure (3.5) presents the performance of the platforms main variables; 1) steering angles values $\dot{\mathbf{q}}_s$, wheel angular velocites $\dot{\mathbf{q}}_x$ (ω), the robot velocities $(\dot{x}, \dot{y}, \dot{\phi})$, and the robot trajectory. The WCE virtually actuates the steering angular velocities to reach the value -45^o, as shown in Figure (3.5-a). Such actuation should occur with at the beginning of the robot motion. Afterwards, the virtual steering actuation will have a negligable effect on the soution and the WCE will yield the J_{in_x} Jacobian (3.2). The effect of this process appears in the translational and rotational velocities as shown in Figure (3.5-c), Figure(3.5-d) and Figure (3.5-e), where they are represented in two frames of coordinates, the C3P frame and the floor frame. As the WCE solution yields to the Jacobian J_{in_x} the oscillations settle to the steady state phase.

Such a result is due to the following: since the WCE is an approximate solution

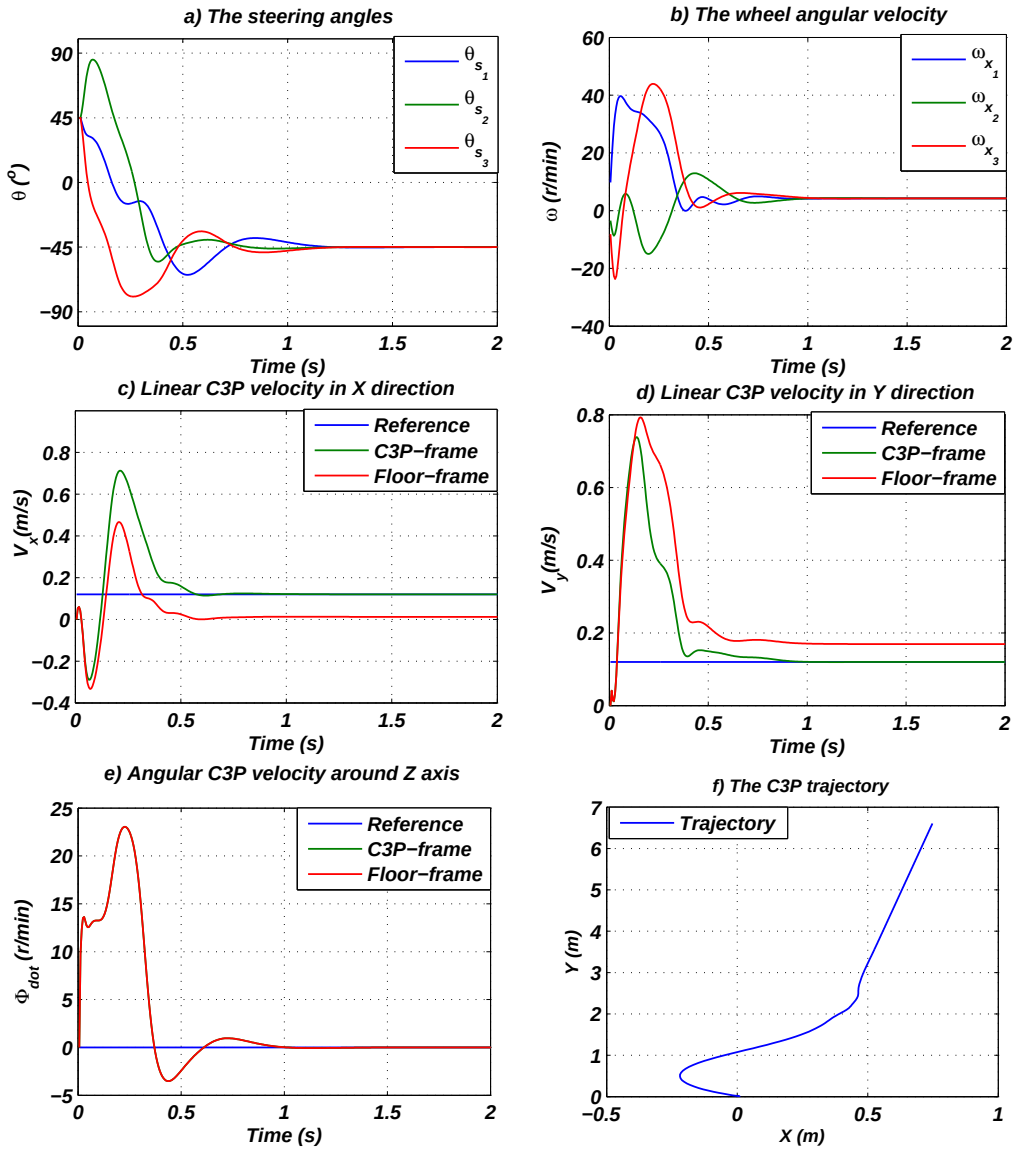

Figure 3.5. The C3P simulation results from example (1); driving in $\dot{\mathbf{p}} = [0.12(m/s)\ 0.12(m/s)\ 0(r/min)]^T$

it generates distortion errors in the robot orientation angle, which result in steady state errors between the C3P and the floor frame velocities. Such errors drive the robot in the wrong direction with respect to the floor coordinates Fig. (3.5-f).

These distortion errors are unpredictable with other initial steering angle values as shown in example 2 Figure(3.6). If the steering angles are all zeros, Figure(3.6-a)

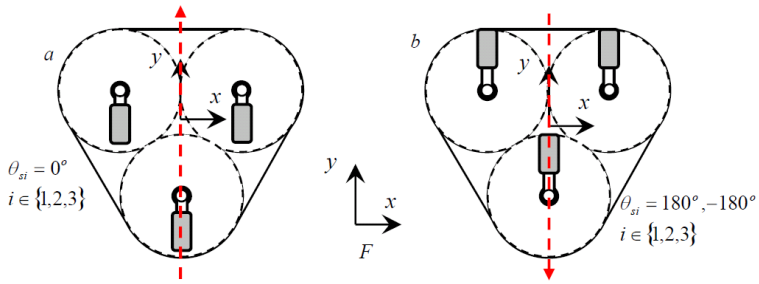

Figure 3.6. C3P wheels configuration considering the results in Figure (3.7)

and the robot is required to drive in a negative y direction with reference velocities $\dot{\mathbf{p}} = [0(m/s) \quad -0.12(m/s) \quad 0(r/min)]$; the steering angles should all rotate to reach the value of -180^o or 180^o (Fig.3.6-b). Such rotation of the steering angles (Fig. (3.7-a)) occurs because of the caster wheel dynamics, which tend to push the robot instead of pulling it. Therefore, the caster wheels will always rotate in the direction of pushing, so that the wheel angular velocities always have positive values. Despite the negative value of the robot desired velocity (Fig. (3.7-c)), the wheel angular velocities in Figure (3.7-b) yield to a positive value.

During the period of the steering angles rotation, the C3P may reach singularity condition or near singularity condition. As a result, the WCE generates wheel velocities' oscillations that settle at t=4 sec when the WCE yields the Jacobian J_{in_x}. The wheel angular velocities also oscillate (Fig.3.7-b), which produce oscillations in the robot velocities (Fig.3.7-c, d & e).

The reason for such behavior is due to the following, the WCE additional term is an approximate solution, which may generate non-unique solutions (the operating points are far from the singularity condition). Therefore the effect of the additional term in the WCE solution should be eliminated as soon as the C3P leaves its singularity condition. A controller that regulates the effect of the WCE additional term must be obtained. Therefore, a method should be developed to measure the singular condition during any experiment.

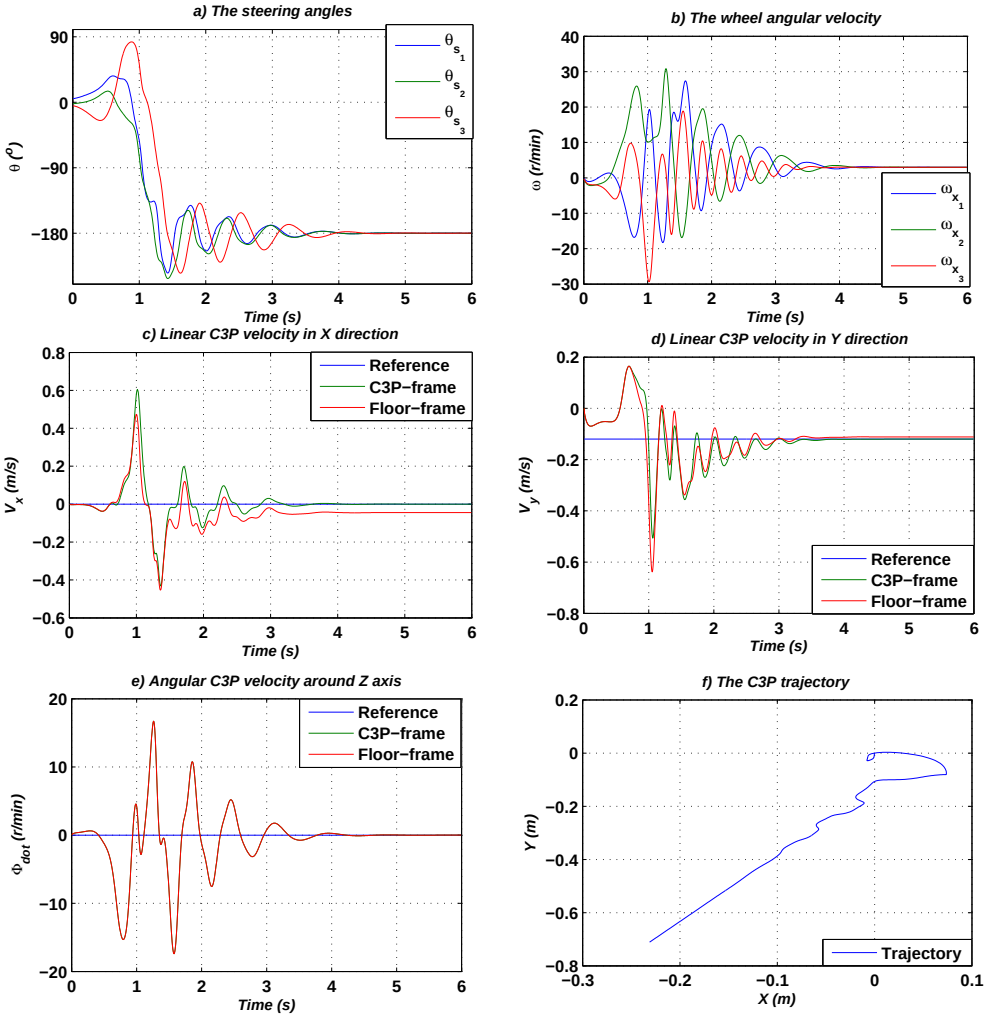

Figure 3.7. The C3P simulation results from example (2); driving in $\dot{\mathbf{p}} = [0(m/s) \ -0.12(m/s) \ 0(r/min)]^T$

3.2.2 Singularity Indicator

This section proposes a geometric method to measure the ability of the C3P to drive in a certain direction. It is clear that the knowledge of the steering angles and the reference robot velocities values are sufficient to detect the singularity of the C3P. For example, if $\theta_{s_i} = 0^o$ and the reference robot velocity is $\dot{\mathbf{p}} = [0 \ 0.12 \ 0]$, then the velocity in y direction is actuated; on the other hand , if $\dot{\mathbf{p}} = [0.12 \ 0 \ 0]$, the velocity in x direction is non-actuated, which is a singular condition.

The singularity indicator $|\Psi|$ will detect the robot singularity conditions. The

angle $\Psi = \delta - \beta$, where $\delta = \arctan(-\dot{y}_a, \dot{x}_a)$ and $\beta = \arctan(-\dot{y}_r, \dot{x}_r)$.

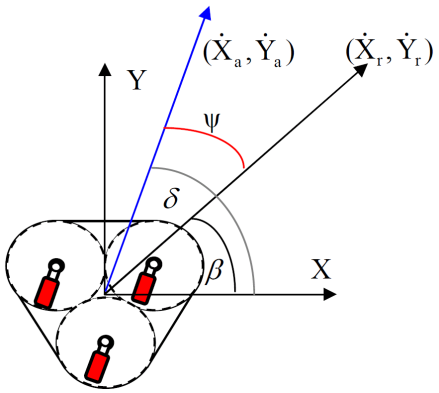

Figure 3.8. The geometric representation for achieving Ψ

$\dot{x}_a, \dot{y}_a$: The measured robot Linear velocities in x and y directions

$\dot{x}_r, \dot{y}_r$: The reference robot Linear velocities in x and y directions

The indicator $|\Psi|$ works as a sensor for the C3P singularity condition. The condition of $|\Psi| = 90^o$ indicates un-actuation in the desired linear velocity direction $(\dot{x}, \dot{y})$. This condition requires full influence from the coupling approach (WCE). And the condition of $|\Psi| = 0^o$ or $|\Psi| = 180^o$ stands for full ability of motion in the desired direction where the coupling approach is not needed.

3.3 Wheel Coupling Equation Adaptation

Equation (3.21) shows that 'm' is the main parameter which has the capability of adapting the effect of the WCE. A mathematical solution for 'm' as a function of the singularity indicator must be found. This function should act as the WCE regulator, which gives the following results. The variable 'm' varies between the values $= [0.5, 0]$; when $|\Psi| = 90^o$ then 'm' takes its maximum value '0.5'. When $|\Psi| = 0^o$ or $|\Psi| = 180^o$, 'm' will be zero to illuminate the influence of the coupling solution. These constraints can be generated from the following equation

$$m = 0.5 \ \sin^{\lambda}(|\Psi|) \tag{3.22}$$

The parameter λ is tuned to regulate the WCE effect on the solution during certain values for Ψ. The WCE regulator is used in the first example shown earlier in this section, where the desired robot velocity is $\dot{\mathbf{p}} = [0 \ \ -0.12 \ 0]$. Figure (3.9) shows the differences in the simulation results with and without using the WCE adapter.

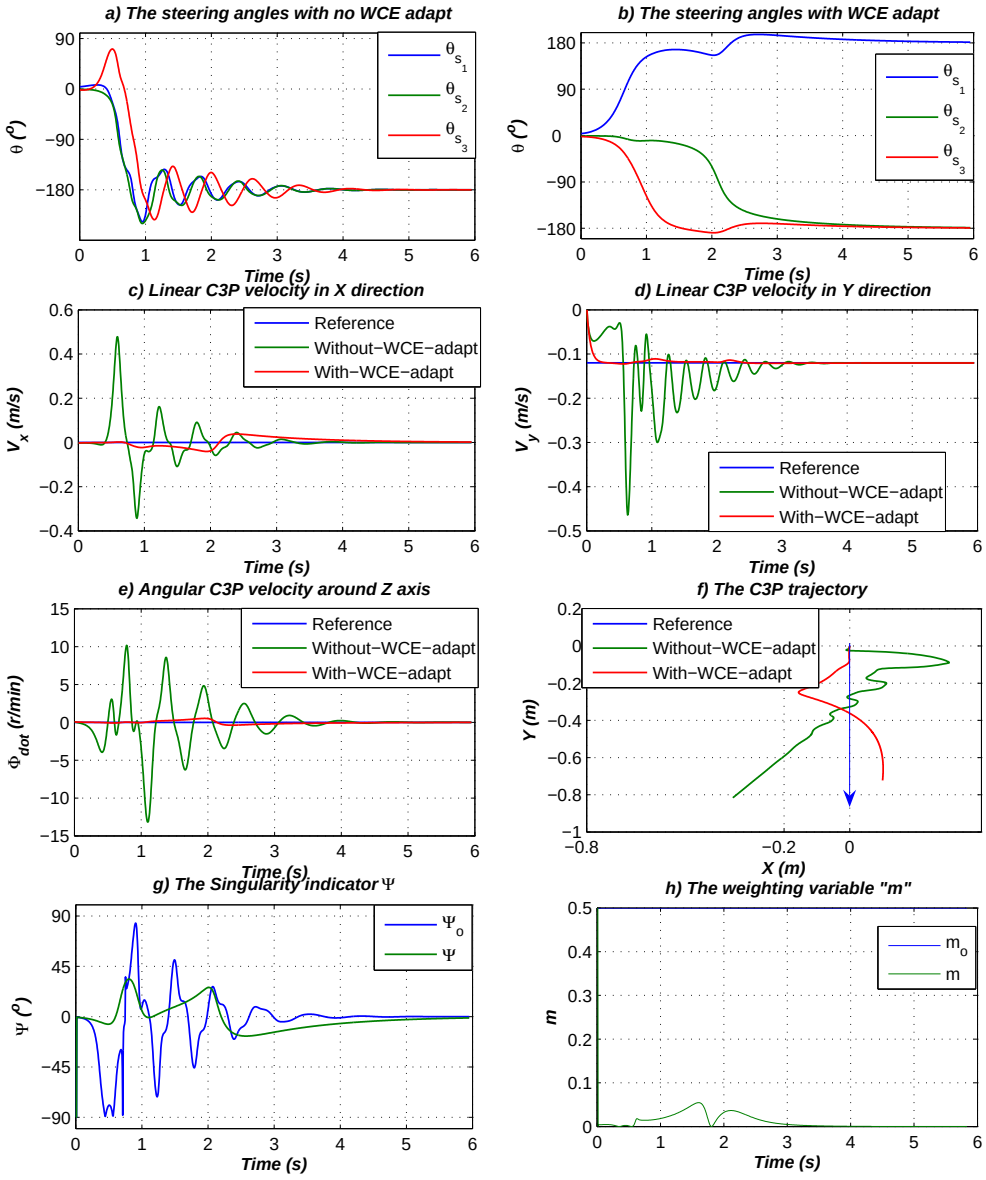

Figure 3.9. C3P simulation results with and without WCE regulator

It is clear from Figure (3.9b) that the oscillations are nearly dampened with

the WCE regulator. The robot velocities are then generated Respectively with almost no oscillations, as shown in Figure (3.9c, d, & e), and if they existed it is only during the steering angles rotation phase. The singularity indicator response shown in Figure (3.9-g) shows that it oscillates rapidly in the case of no adaptation for the WCE.

When the WCE adapter is introduced in the solution, the parameter 'm' takes zero value instead of 0.5 to reduce the influence of the additional coupling term in the WCE. Figure(3.9-h) shows that during the wheels steering rotation, the parameter 'm' oscillates to give the coupling term the opportunity to maintain reference velocities. The C3P trajectories (Fig. 3.9f) shows that by using the WCE adapter, the robot drives in a direction close to the desired direction (-y), but with small errors. The next step is to obtain a controller to control the robot velocities after the adjusting steering angles phase.

3.4 Velocity Controller

In this section, a velocity controller with the following specifications must be designed a) adjust the steering angles to the desired value, b) adjust the robot coordinates to match the floor coordinates without influencing the steering angles, c) the output robot velocities follows the input signals corresssponding to the floor frame of coordinates. For a holonomic WMR, a regular PID controller is suitable for controlling the robot velocities.

For the C3P actuators configuration however, the PID controller alone cannot manage the three tasks. The PID controller can manage the second and third specifications and the WCE regulator manage the first specification. If both controllers are used in parallel, the PID controller conflicts with the coupling approach solution. Such a conflict will cause unstable behaviour of the robot. Therefore the PID controller should occur after the adjustment of the steering angles.

The controller's structure is shown in Figure (3.10), where switching is used between the WCE regulator and the PID controller. The switching conditions are shown in Table (3.2), where ϵ is a tolerance value of the singularity indicator $|\Psi|$. The PID controller is described with the following equations:

Table 3.2. The controller xondition

Indication $\|\psi\|$	Condition 1	Condition 2
$\|\psi\| < \epsilon$	True	False
$\|\psi\| > \epsilon$	False	True

$$\begin{aligned} PID_x &= K_{P_x} + K_{I_x}/s + K_{D_x}s \\ PID_y &= K_{P_y} + K_{I_y}/s + K_{D_y}s \\ PID_\phi &= K_{P_\phi} + K_{I_\phi}/s + K_{D_\phi}s \end{aligned} \tag{3.23}$$

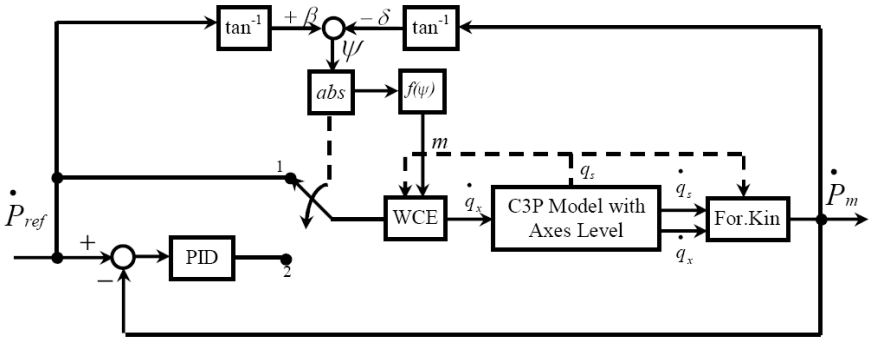

Figure 3.10. C3P velocity controller structure

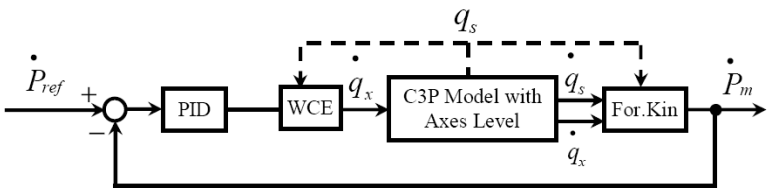

Figure 3.11. C3P velocity controller structure

To show the difference between using the PID controller alone and combining it with the WCE adapter, the same previous example is carried out with the control structure shown in Figure (3.10) and again with the structure in Figure (3.11). The structure in Figure (3.10) has the following controller parameters: $\lambda = 2$,

$\epsilon = 5^o$, $K_{P_x} = 0.1$, $K_{P_y} = 0.1$, $K_{P_\phi} = 0.2$ and $K_{D_\phi} = 0.15$. On the other hand, the PID controller in Figure (3.11) has the following parameters: $K_{P_x} = K_{P_y} = 0.1$, $K_{I_x} = K_{I_y} = 0.05$, $K_{P_\phi} = 0.25$ and $K_{I_\phi} = 0.15$. Figure (3.12) shows the deficiency of using the only PID controller.

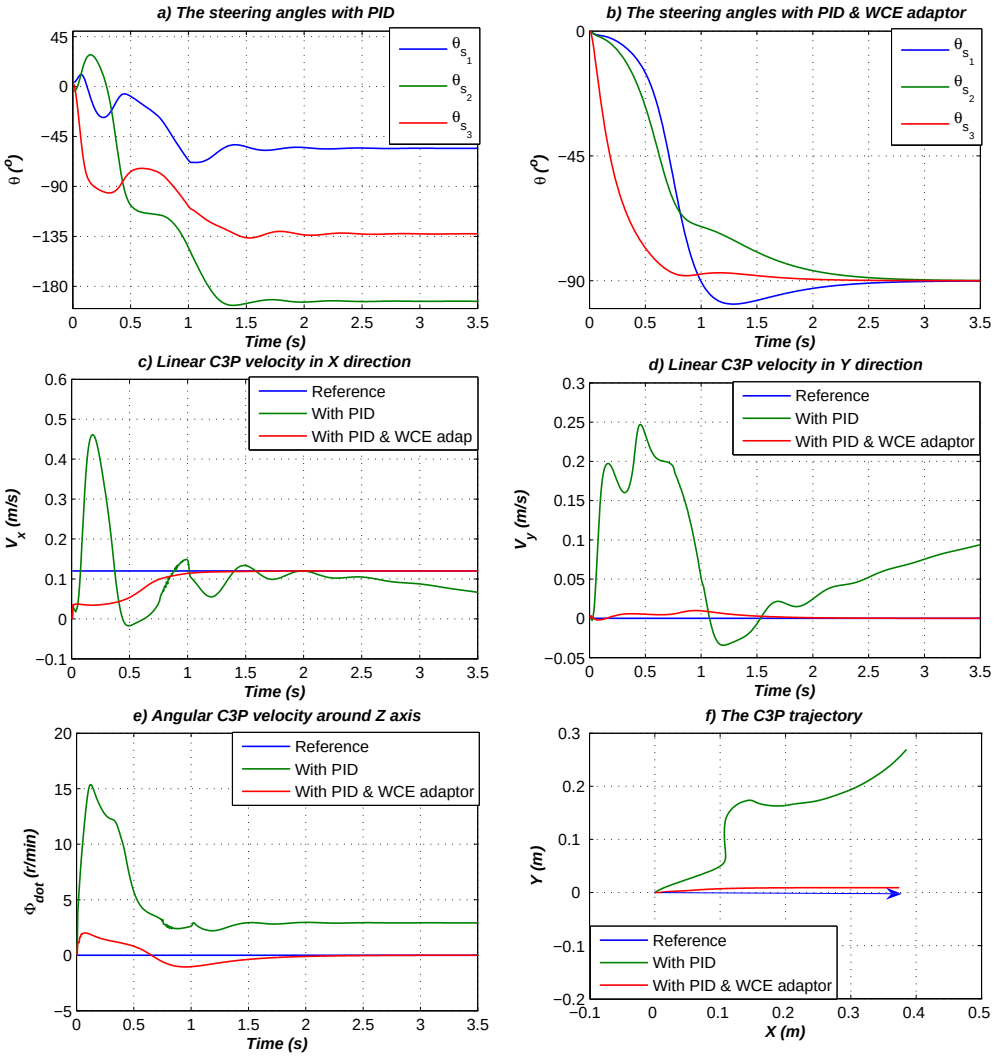

Figure 3.12. C3P performance with and without the velocity controller

First, it should be taken in consideration that the robot velocity responses are referred to the floor frame of coordinates. Figure (3.12-a & 3.12-b) show that using the PID controller alone was insufficient to settle the steering angles to the

value of -90^o, while the proposed structure in Figure (3.11) achieved the expected steering angles values. This is noticed from the robot velocity responses (Fig. 3.12 c & d), where the WCE and its adapter adjust the steering angles to reach the desired value -90^o. Then the velocity controller succeeded to reach the desired robot velocities with respect to the floor coordinates.

The WCE reduces the oscillations found in the angular robot velocity response (Fig. 3.12 e). The robot trajectory in Figure (3.12 f) shows the efficient performance of the controller platform to drive in the exact desired direction.

3.5 Position Controller

The position control problem of Wheeled Mobile Robots (WMRs) has been a heavily researched area due to both the challenging theoretical nature of the problem and its practical importance. Researchers have targeted the problems of: regulating the position/orientation of the WMR, and the tracking time-varying reference trajectory .

The position control problem can be stated as follows: Given the position goal coordinates vector $\dot{\mathbf{p}}_g = \begin{bmatrix} x_g & y_g & \phi_g \end{bmatrix}^T$ and the initial position reference coordinates vector $\dot{\mathbf{p}}_g = \begin{bmatrix} x_i & y_i & \phi_i \end{bmatrix}^T$. It is necessary to design a state feed back controller if it exists, such that the control $u = K.e = \begin{bmatrix} U_x & U_y & U_\phi \end{bmatrix}^T$ makes $lim_{t\to\infty} e(t) = 0$. The robot kinematics descriptions are given by the following:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\phi} & 0 \\ \dot{\phi} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \phi_e \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_x \\ U_y \\ U_\phi \end{bmatrix} \tag{3.24}$$

The state space form is written in the form of $\dot{\mathbf{p}} = Ap + Bu$, where $\mathbf{p} = \mathbf{p}_g - \mathbf{p}_i = [x_e \; y_e \; \phi_e]^T$. Here we have three controlled states $\dot{x}$, $\dot{y}$ and $\dot{\phi}$. There is a fourth sensed state; it is an input sensed variable to control the angular robot velocity. This sensed variable is $\delta_{er} = \vartheta - \delta_g$ where $\vartheta = \arctan(\frac{-Y_g}{X_g})$ and $\delta_g = \arctan(\frac{-Y_e}{X_e})$ (Fig.3.13)

The velocity controller succeeded in driving the robot in the floor coordinates

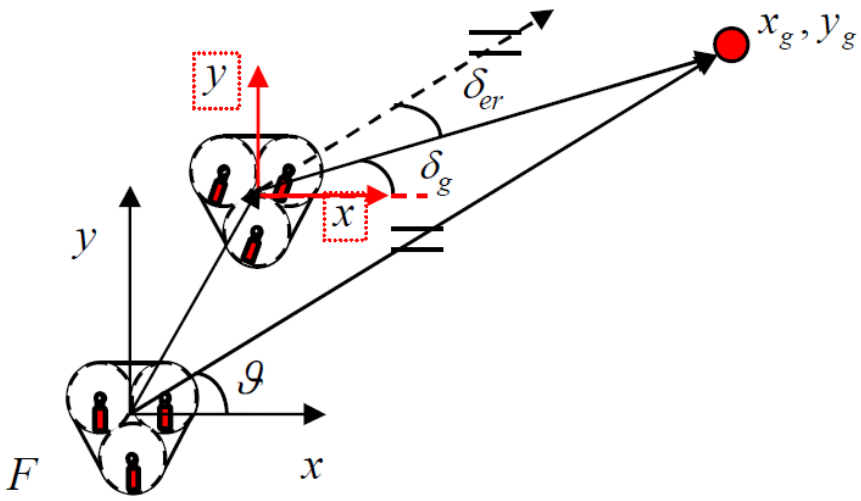

Figure 3.13. Robot Position representation

but aimed for the wrong goal point due to a distance error. The angle δ_{er} is the difference angle between the direction of the robot and the error vector at every instant. The controller will be a non linear static state-feedback, which is represented below:

$$U_x = K_x x_e \tag{3.25}$$

$$U_y = K_y y_e \tag{3.26}$$

$$U_\phi = K_\phi \phi_e + K_{er} \delta_{er} \tag{3.27}$$

Then the closed loop control system can be formulated as follows:

$$\dot{x} = (K_\phi \Phi_e) y_e + K_x x_e \tag{3.28}$$

$$\dot{y} = -(K_\phi \Phi_e) x_e + K_y y_e \tag{3.29}$$

$$\dot{\phi} = +K_\phi \phi_e + K_{er} \delta_{er} \tag{3.30}$$

In order to verify the proposed controller, several simulation examples will take place. The first example is a singular initial condition. The initial point coordinates are $\mathbf{p}_i = [0\ 0\ 0]^T$ starting from singularity wheel configuration $\theta_{s_i} = 0$, $i \in \{1, 2, 3\}$ and the goal point coordinates are $\mathbf{p}_g = [4\ 0\ 0]^T$. The main effective

control variables for smooth trajectory are Φ_e and δ_{er}. In Figure (3.14) the effect of δ_{er} is shown, it is noticed that as the value of K_{er} increases, the robot trajectory errors are aslo reduced from 0.4 m to 0.1 m.

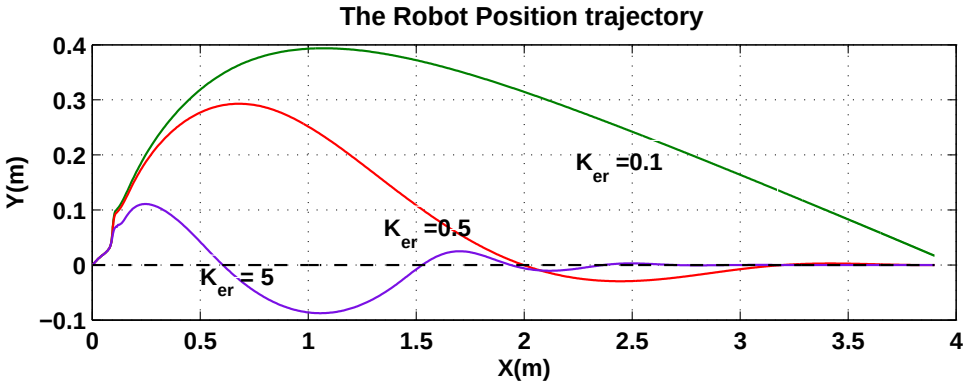

Figure 3.14. Robot trajectory for $K_\phi = 20$ and different K_{er}

Figure (3.15) represents the different robot trajectories produced from different K_ϕ values. When the robot starts moving from the singularity configuration, the kinematics adjust the steering angles to the desired value. As a result, a displacement error is generated, the main function of K_ϕ and K_{er} is to overcome this error smoothly.

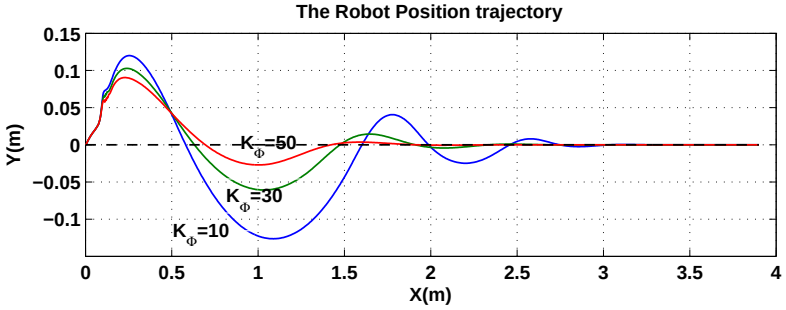

Figure 3.15. Robot trajectory for $\delta_{er} = 5$ and differentΦ_e

The following example requires the C3P to move from $\mathbf{p}_i = [0\ 0\ 0]^T$ towards $\mathbf{p}_g = [2.8m\ 0\ 0]^T$ along the x axis from the zero initial angles. The position controller has the parameters of $K_\phi = 50$ and $K_{er} = 5$. Figure (3.16) shows the robot position update trajectory visualized with the steering angles and robot orientation, which illustrates the performance of the position controller.

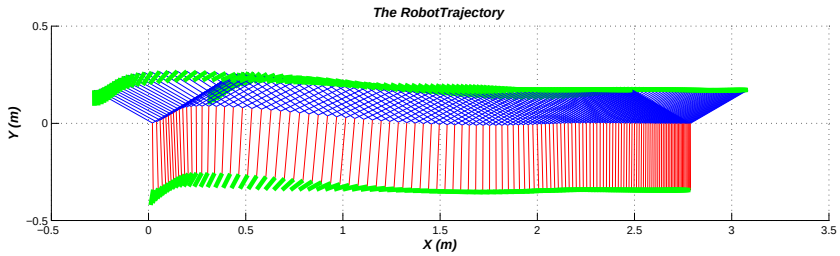

Figure 3.16. Robot Position update

The simulation results for the C3P steering angles and velocities are shown clearly in Figure (3.17). The C3P starts from a singularity condition where $\theta_{s_i} = 0$, $i \in \{1, 2, 3\}$ and is desired to reach the goal coordinate $\mathbf{p}_g$, which is driving in the x direction. From the robot velocities Figures (3.17c & d), the WCE solution and its regulator succeeded in adjusting the steering angles to the near desired value -90^o after two and half seconds; as a result the velocity controller is initiated to drive the robot to the desired direction with respect to the floor frame of coordinates (Fig. 3.17f). Figure (3.18) shows the robot trajectory for a different example, where $\mathbf{p}_i = [0\ 0\ 0]^T$ towards $\mathbf{p}_g = [3m\ 3m\ -90^o]^T$. The robot rotated nearly 90^0 in a clockwise direction and adjusted its steering angles in the same period of time and in displacement of 15 cm; then the robot drove towards the goal coordinates on the desired trajectory.

3.6 Summary

The 'Wheel Coupling Equation' (WCE) is proposed as an actuation inverse solution. The WCE actuates the steering angular velocities by producing distortions of the exact solution in the proximity of singularities. This leads to an escape from the singularity. If the wheel configuration is far from the singular condition, the previous solution is valid and the approximate solution influence will disappear. The

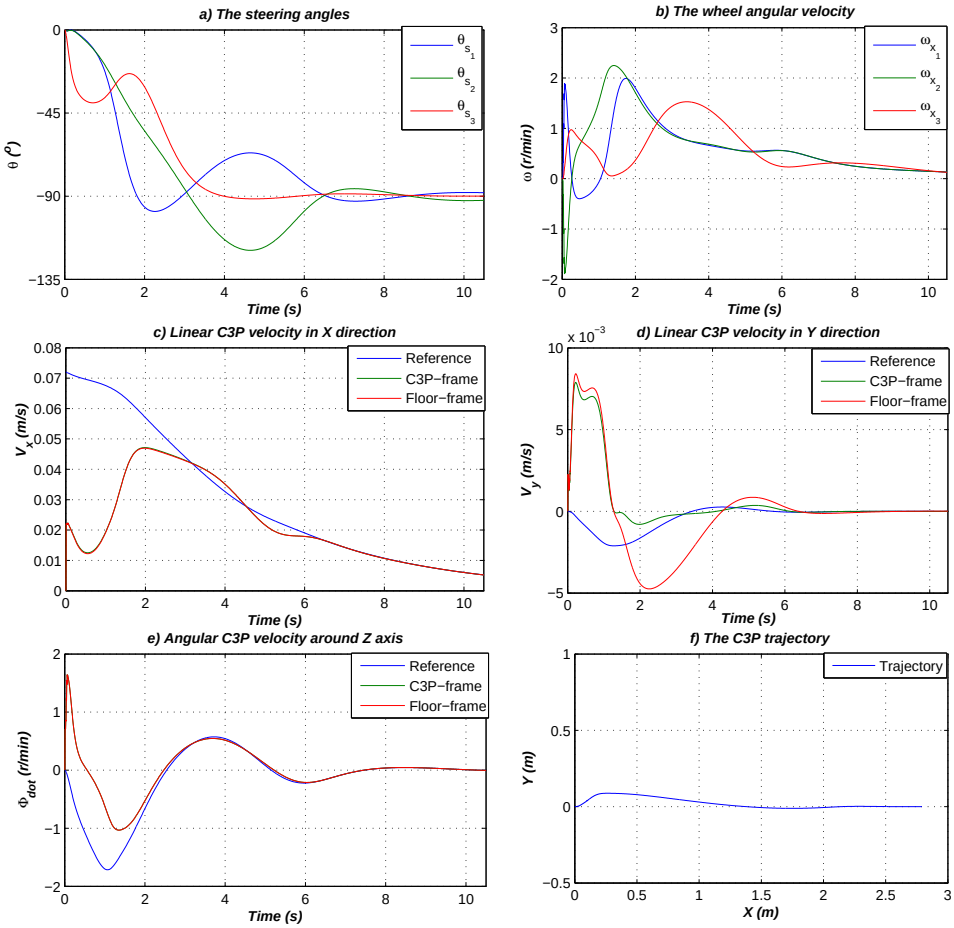

Figure 3.17. Simulation result for driving from $\mathbf{p}_i = [0\ 0\ 0^o]^T$ to $\mathbf{p}_g = [3\ 3\ -90^o]^T$

velocity controller structure consists of two main controllers: The WCE regulator to eliminate the oscillations created by its approximation, and the PID controller to maintain the C3P reference speed with respect to the floor coordinates.

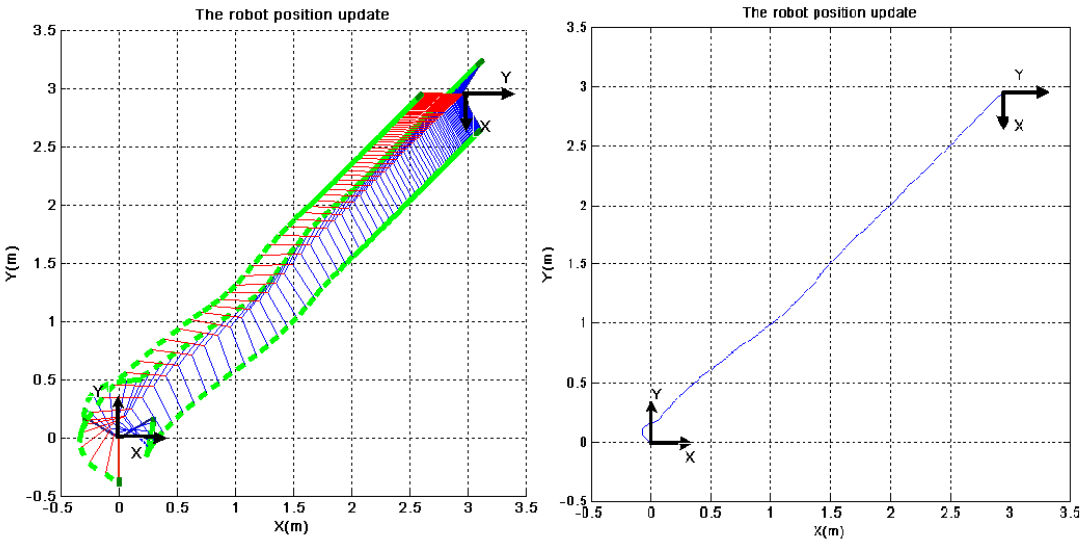

Figure 3.18. Position update and trajectory for $\mathbf{p}_g = (3\ 3\ -90^o)^T$

Chapter 4

Inverse Dynamics Based Motion Control and Analysis

The kinematics based controller (KBC) proposed in the previous chapter succeeded in solving the C3P singularity problem. However, the proposed solution is still an approximate solution, which is developed under the assumptions mentioned in Section 3.2. These assumptions can only be considered for the kinematic modeling and cannot be ignored in the dynamic modeling. Therefore, the inverse dynamic solution is needed to solve the singularity problem accurately.

The inverse dynamic solution determines the forces subjected by the system, for both the dynamic and the kinematic models. The inverse dynamic solution yields the driving forces necessary to control a system so that it follows a desired trajectory [87] [88] and [89].

In this chapter an inverse dynamics solution is proposed in Section 4.1 using the Lagrangian method. In Section 4.2 the velocity and position controllers are presented, along with some simulation examples to illustrate their performance.

4.1 Inverse Dynamics Solution

The solution's main objective is to create an inverse dynamic model based on the sensed variables ($\dot{\mathbf{q}}_x$ and $\dot{\mathbf{q}}_s$) with respect to the actuated variables ($\dot{\mathbf{q}}_x$). Firstly, the robot translational and rotational velocities are decomposed from equation (2.15) to obtain the platform kinetic energy

$$K_p = \frac{1}{2}\, m_p\ \boldsymbol{V}^T_{p(DSE)}\ \boldsymbol{V}_{p(DSE)} + \frac{1}{2}\, \Omega^T_{p(DSE)}\ I_p\ \Omega_{p(DSE)} \tag{4.1}$$

where m_p is the wheel mass and I_p is the wheel inertia located at the platform center of gravity. Then the Lagrangian function $L = K_p = f(\mathbf{q}_s, \dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s)$ is used to obtain the following Lagrangian formulation

$$\tau_{x_a} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_x}\right) - \frac{\partial L}{\partial \mathbf{q}_x}. \tag{4.2}$$

where the generalized velocities are $\mathbf{q}_g = \mathbf{q}_x$. The Lagranigian fomulation will yield the following torque equation

$$\tau_{x_a} = M_{sx_a}\ddot{\mathbf{q}}_{sx} + G_{sx_a}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{4.3}$$

The matrix M_{sx_a} is decomposed into M_{x_a} and M_{s_a}, where $M_{x_a}, M_{s_a} \in R^{3X3}$

$$M_{sx_a} = \left[\ M_{x_a}\ M_{s_a}\ \right], \quad \ddot{\mathbf{q}}_{sx} = \begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{bmatrix} \tag{4.4}$$

where M_{xa} is a mass matrix corresponding to the wheel angular acceleration vector $\ddot{\mathbf{q}}_x$ and M_{sa} corresponds to the steering angular acceleration vector $\ddot{\mathbf{q}}_s$, while the centripetal and Coriolis torques are presented by the vector G_{sx_a} (Appendix C).

The dynamic torque equation (4.4) is a function of $\ddot{\mathbf{q}}_x$, $\ddot{\mathbf{q}}_s$, $\dot{\mathbf{q}}_x$. On the other hand, the required inverse dynamic solution should have the robot velocities $\dot{\mathbf{p}}$ and accelerations $\ddot{\mathbf{p}}$ as inputs.

Corresponding to the caster wheel, the forward kinematic solution for the wheel accelerations is

$$\ddot{\mathbf{p}} = J_{f_i}\ddot{\mathbf{q}}_i + J_{r_i}\dot{\mathbf{q}}_{r_i} \tag{4.5}$$

where J_{f_i} is the forward direct solution for the wheel accelerations, and $J_{r_i}\dot{\mathbf{q}}_{r_i}$ is the Centripetal and Coriolis velocities.

$$J_{f_i} = \begin{bmatrix} -r\sin(\theta_{s_i}) & h\cos(\alpha_i) + d\sin(\theta_{s_i}) & -h\cos(\alpha_i) \\ r\cos(\theta_{s_i}) & -h\sin(\alpha_i) + d\cos(\theta_{s_i}) & h\sin(\alpha_i) \\ 0 & 1 & -1 \end{bmatrix},\ \dot{\mathbf{q}}_{r_i} = \begin{bmatrix} \dot{\theta}^2_{s_i} \\ -2\dot{\theta}_{c_i}\dot{\theta}_{s_i} \\ \dot{\theta}^2_{c_i} \end{bmatrix} \tag{4.6}$$

and

$$J_{r_i} = \begin{bmatrix} d_{sx_i} & d_{sx_i} & d_{x_i} \\ d_{sy_i} & d_{sy_i} & d_{y_i} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} h\sin(\alpha_i) & h\sin(\alpha_i) & h\sin(\alpha_i) - d\cos(\theta_{s_i}) \\ h\sin(\alpha_i) & h\cos(\alpha_i) & h\sin(\alpha_i) + d\cos(\theta_{s_i}) \\ 0 & 0 & 0 \end{bmatrix}. \quad (4.7)$$

The inverse actuated kinematic accelerations are

$$\begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{bmatrix} = J_{in_v}\ddot{\mathbf{p}} - g_{cs}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_c) \quad (4.8)$$

where

$$J_{in_v} = \begin{bmatrix} J_{in_x} \\ J_{in_s} \end{bmatrix} \quad (4.9)$$

and the vector g_{cs} is explained in detail in Appendix (C).

The inverse kinematic velocities solutions for J_{in_x}(2.10) and J_{in_s}(2.12), along with inverse acceleration kinematics solution in (4.6) and (4.8) are used to reformulate the torque equation (4.4) to deliver the following torque equation

$$\tau_{x_a} = M_x(\mathbf{q}_s)\ddot{\mathbf{p}} + G_x(\mathbf{q}_s, \dot{\mathbf{p}}), \quad (4.10)$$

which is function of the robot velocities $\dot{\mathbf{p}}$ and accelerations $\ddot{\mathbf{p}}$, in addition to

$$M_x = M_{sx_a}J_{in_v}, \quad and \quad G_x = G_{sx_a}(\mathbf{q}_s, \dot{\mathbf{p}}) - M_{sx_a}g_{cs}(\mathbf{q}_s, \dot{\mathbf{p}}). \quad (4.11)$$

4.2 Dynamics Based Motion Control Structure

The Dynamic Based Controller (DBC) consists of two main loops the velocity control loop [90] and the position control loop [91]. In this section, the inverse dynamic solution is used within the velocity and position controller structure as shown in Figure(4.1) and Figure (4.2). The Axes level control loop is applied to the torque control of each wheel (T.Ctrl). The velocity controller (vel.Ctrl.) is used to calculate the required error in the robot accelerations ($\ddot{\mathbf{e}}$), which are added to the reference robot acceleration $\ddot{\mathbf{p}}_r$ (Fig.4.1). The actuated wheels torques τ_{xa} are generated from the inverse dynamic solution using the reference robot velocities

$\dot{\mathbf{p}}_r$ and accelerations $\ddot{\mathbf{p}}_r + \ddot{\mathbf{e}}$ as inputs. The position controller is used to calculate the reference velocity vector $\dot{\mathbf{p}}_r$ needed to drive the robot to goal coordinates $\mathbf{p}_g$ (Fig4.2). The control structure presented in this section is illustrated by simulation examples, which are carried out with respect to the floor frame of coordinates.

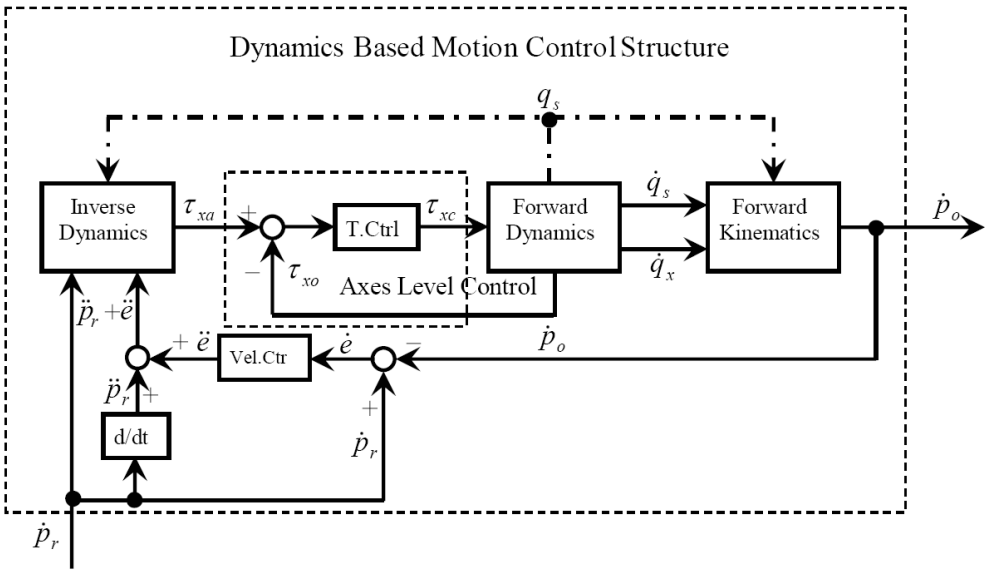

Figure 4.1. Dynamics Based Velocity Control Structure

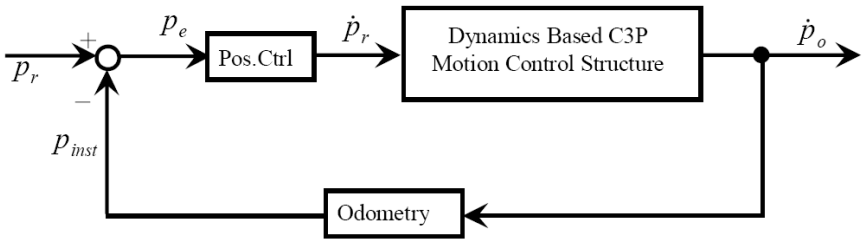

Figure 4.2. Position Control Structure

4.2.1 Velocity Controller

The velocity controller of the inverse dynamic solution (Vel.Ctrl) controls the robot accelerations vector $\ddot{\mathbf{p}}_r$, which is one of the main factors influencing the solution. Its structure is described through the following equation

$$\ddot{\mathbf{p}}_r = \Gamma_{V.C}(S)\dot{\mathbf{p}}_e$$
$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} C_x & 0 & 0 \\ 0 & C_y & 0 \\ 0 & 0 & C_\phi \end{bmatrix} \begin{bmatrix} \dot{x}_r - \dot{x}_o \\ \dot{y}_r - \dot{y}_o \\ \dot{\phi}_r - \dot{\phi}_o \end{bmatrix} \tag{4.12}$$

where, $\Gamma_{V.C}$ is the velocity control matrix. It is a diagonal matrix with the PID controller for linear acceleration in x direction

$$C_x = K_{px} + K_{dx}S + \frac{K_{ix}}{S}, \tag{4.13}$$

and in y direction

$$C_y = K_{py} + K_{dy}S + \frac{K_{iy}}{S} \tag{4.14}$$

and for the angular robot velocity $\ddot{\phi}$

$$C_\phi = K_{p\phi} + K_{d\phi}S + \frac{K_{i\phi}}{S} \tag{4.15}$$

4.2.2 Dynamics Performace Examples

This section shows the performance of the inverse dynamics with the aid of some examples. The parameters used throughout this section are represented in Table. (4.1). The following simulation example is used to demonstrate the effect of the wheel variables velocities and accelerations ($\dot{\theta}_s$=ω_s , $\ddot{\theta}_s$=α_s, $\dot{\theta}_x$=ω_x and $\ddot{\theta}_x$=α_x) on the C3P behavior.

Table 4.1. The C3P Control parameters

C3P Parameters	Value	Units	Ctrl.Par	Value
h	0.343	m	$K_{px} = K_{py}$	1.1
d	0.04	m	$K_{ix} = K_{iy}$	0.9
r	0.04	m	$K_{dx} = K_{dy}$	0
M_p (P_l mass)	30	Kg	$K_{p\phi}$	1.1
I_p (P_l inertia)	3.51	$Kg\ m^2$	$K_{d\phi}, K_{i\phi}$	1.2,0.2

The robot accelerations inputs are $\ddot{p}_r = [0.0035(m/s^2)\ 0(m/s^2)\ 0(r/min^2)]^T$, which is a velociy input in the x direction. The initial steering angles values are

zeros $\mathbf{q}_s = [0^o\ 0^o\ 0^o]^T$(Fig.4.3-a). Such an input yields the steering angles to reach value of -90^o (Fig.4.3-b).

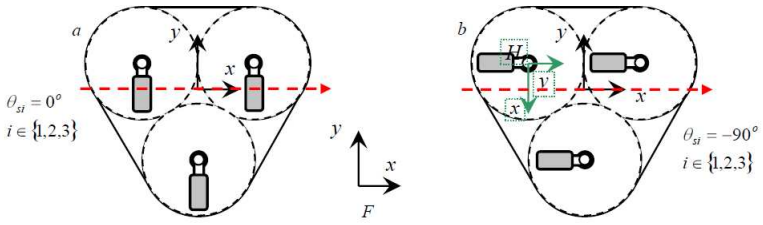

Figure 4.3. The Steering angles orientation for driving in X direction from configuration (a) to configuration (b)

Figure(4.4) shows the steering angle trajectories in addition to the robot and wheel velocities and accelerations. The inverse dynamics solution produces the required torque needed to adjust the steering angles to the direction of motion. Such torque values produced the wheel angular velocities and acceleration responces shown in Figures (4.4d & e), which virtualy actuated the steering angular velocities and acceleration shown in (Fig.4.4b & c). These signals are fed to the C3P model for actuating the steering angles to reach the desired value -90^o or 270^o. The velocity controller succeeds in controlling the robot and accelerations, as shown in Figure(4.5).

The robot will drive in x direction with a ramp input velocity (step input acceleration) untill it reaches $0.12(m/s)$ and then it will maintain a constant speed (zero acceleration). As shown in Figure(4.5a), the measured velocity in x direction oscillates around the reference slope before it settles to a steady state. The measured acceleration in x direction oscillates for 6 seconds before it reaches stability, as shown in Figure (4.5b). Similar responses are noticed in the y direction velocities and accelerations (Fig.4.5c & d).

The inverse kinematics and inverse dynamics based controller are used in the next example to compare their performance. The example represents a singular condition movement which was shown in Figure(4.3). Where the initial steering angles values are $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = 0^o$ and the reference robot velocity $\dot{\mathbf{p}}_r = [0.12(m/s)\ \ 0(m/s)\ \ 0(r/min)]^T$. This input actuates the steering angles to reach the value of -90^o.

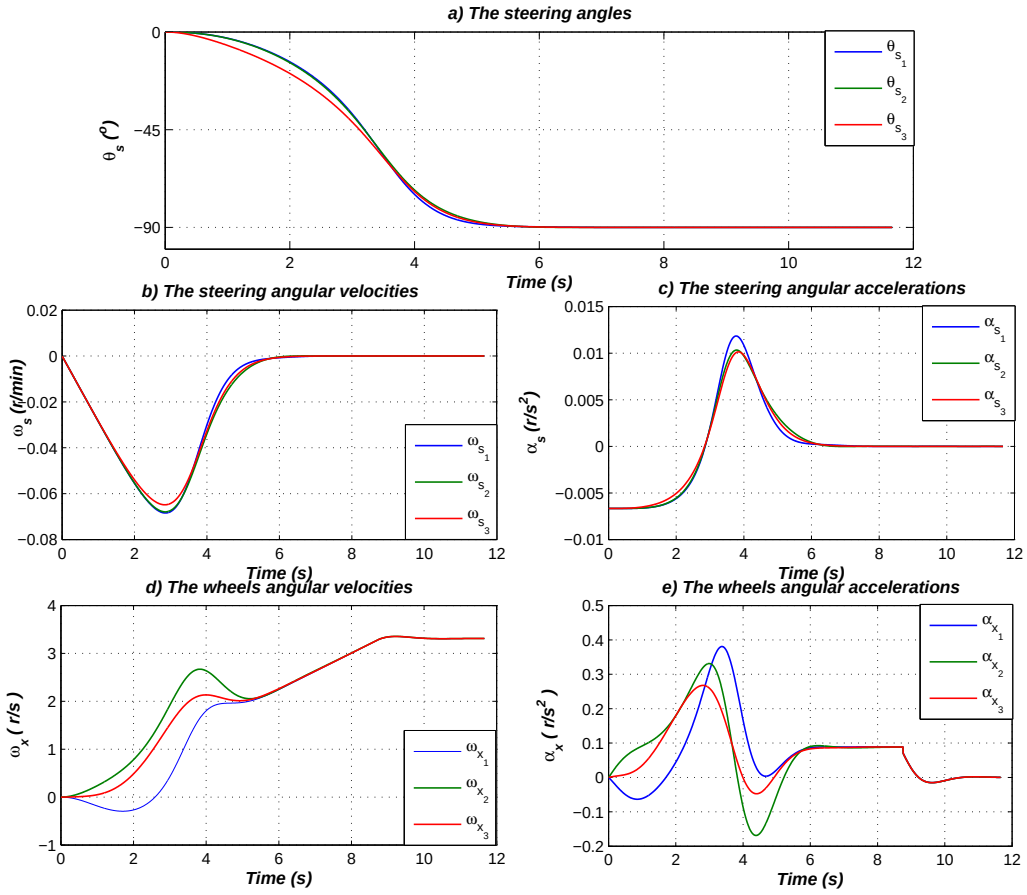

Figure 4.4. The wheels velocities and acceleration for ramp input and driving from singular condition

The steering angles were a result to the dynamic solutions shown in Figure(4.6-b) have small differences in their trajectories during the transition phase. The steering angles reach the desired value (-90^o) almost during the same time interval when time= 1 s. The inverse kinematic solution (WCE) is an approximate solution, which generates non similar values for the steering angles values during the transient period (Fig.4.6a). The angles reach the desired value in different timing, for example: θ_{s_3} reaches -90^o in one second while θ_{s_1} and θ_{s_2} settle at the desired value after two seconds, with difference in values. The advantages of the proposed dynamic solution are noticeable in the robot velocity responses. In Figure (4.6-c), the linear velocity $\dot{x}$ or V_x reaches the reference value faster and smoother than the one resulted from the inverse kinematic solution. The linear

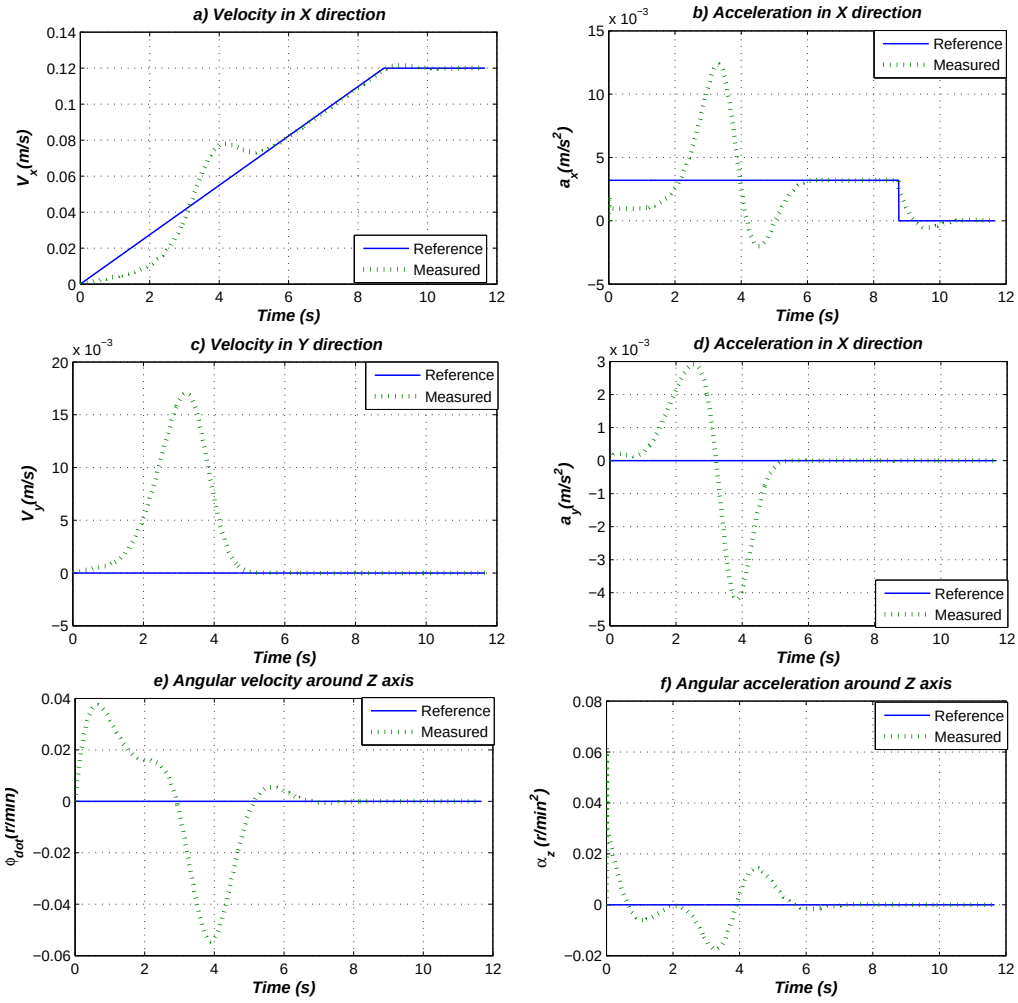

Figure 4.5. The robot velocities and acceleration for ramp input and driving from singular condition

velocity $\dot{y}$ or V_y generated by the inverse kinematic solution has two high overshoots for more than two and half seconds before it reaches the reference value at Time= 4 s (Fig.4.6-d). The V_y generated by the inverse dynamic solution carries a single overshoot for less than one second. Then it reaches the reference value and the steady state in only one second. Such vast differences in theas V_y responses eliminate the expected robot displacement errors in the y direction.

The output $\dot{\phi}$ (Fig.4.6-e) of the inverse dynamic solution has only one overshoot with one quarter of the magnitude of the positive overshoot resulted from the

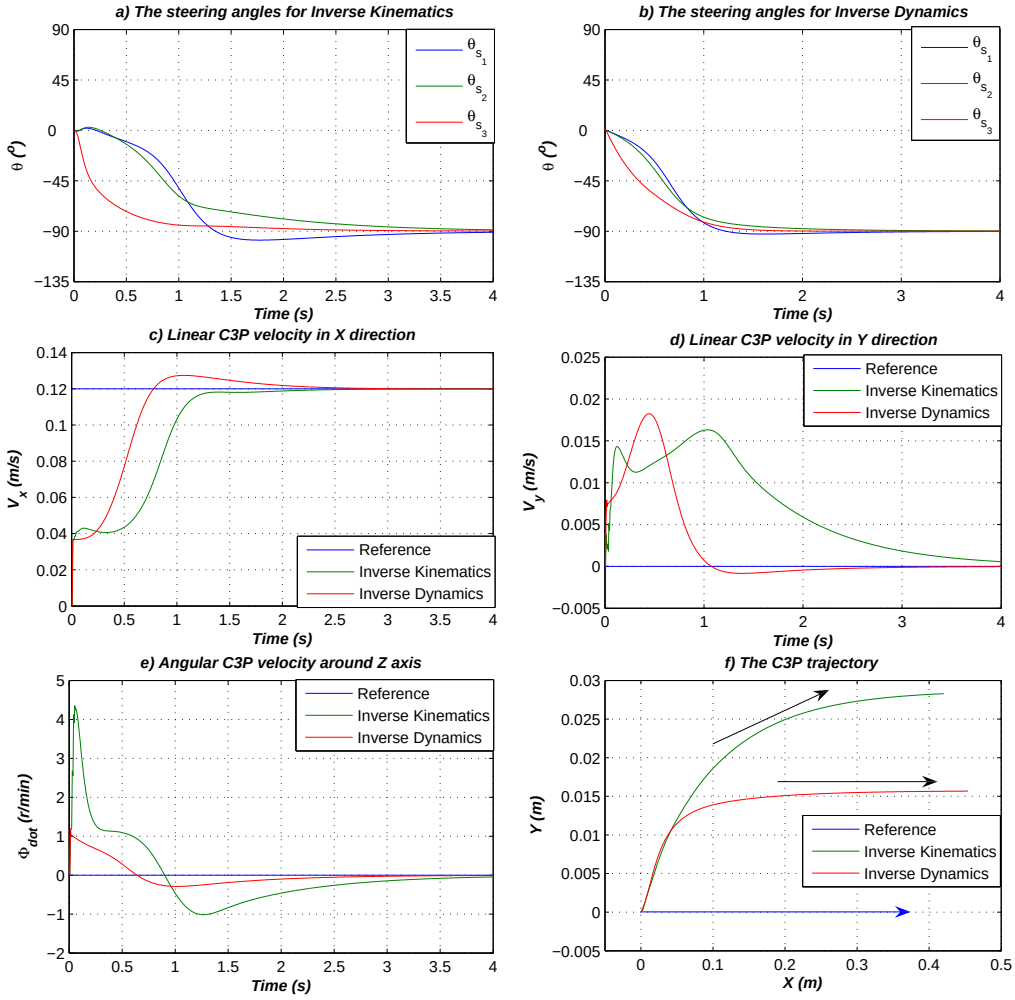

Figure 4.6. Comparing dynamic and kinematic inverse solutions for deriving in x direction from initial singular condition

kinematic solution. Furthermore, the orientation velocity of the dynamic solution settles after one and half seconds while the kinematic solution settles its orientation velocity after almost three seconds. Figure (4.6-f) shows the integration of the robot velocities, which shows the high effect of their errors on the robot trajectories. The robot inverse dynamic solution trajectory takes smooth slope in x, y direction during the transient phase untill the steering angles are adjusted to a -90^o value, while the trajectory resulted from the inverse kinematic solutions takes longer to reach the desired direction. The displacement error found in the y direction exists

for both solutions, however, the kinematics error is double the dynamics one.

The third example is also a comparison between the two solutions but it is set to drive in the y direction using the same controller parameters. The initial steering angles values are $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = -90^o$ as shown in Fig. (4.7-a) and the reference robot velocity $\dot{\mathbf{p}}_r = [0(m/s) \;\; 0.12(m/s) \;\; 0(r/min)]^T$. For such input the steering angles should reach the value of 0^o (Fig. 4.7-b)

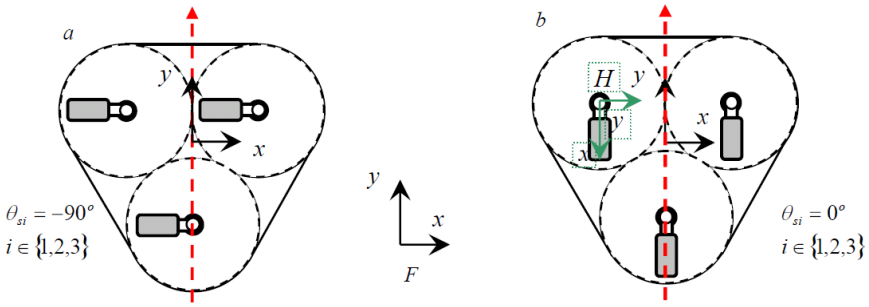

Figure 4.7. The steering angles orientation for driving in y direction from configuration (a) to configuration (b)

From Figure (4.8) the inverse dynamics solution generates a smooth trajectory for the steering angles. On the other hand, oscillations are found in the transient phase for the ones generated by the inverse kinematic solution.

Such differences in trajectories influence the robot linear and angular velocities (Fig.4.8c,d & e). The velocities resulted from the dynamic based controller (DBC) has a very small overshoot with a short settling time in comparison to the kinematics based controller (KBC). As shown in Figure (4.8f) the displacement errors on x axis, generated by the DBC, are half the displacement errors generated from the KBC. As a conclusion: the DBC perform much better than the KBC for the following reasons: a) the wheel angular velocities results better trajectories for the steering angles, b) the robot velocities overshoots are less oscillatory and reduced in magnitude, c) the displacement errors does not exceed the half values resulted from the inverse kinematics solutions.

The next step is to use the torques (τ_x) resulting from the inverse dynamics solution as a feed forward signal parallel to the inverse kinmatics solution. This is presented in the 'Dynamics and Kinematics fusion' block shown in Figure (4.9).

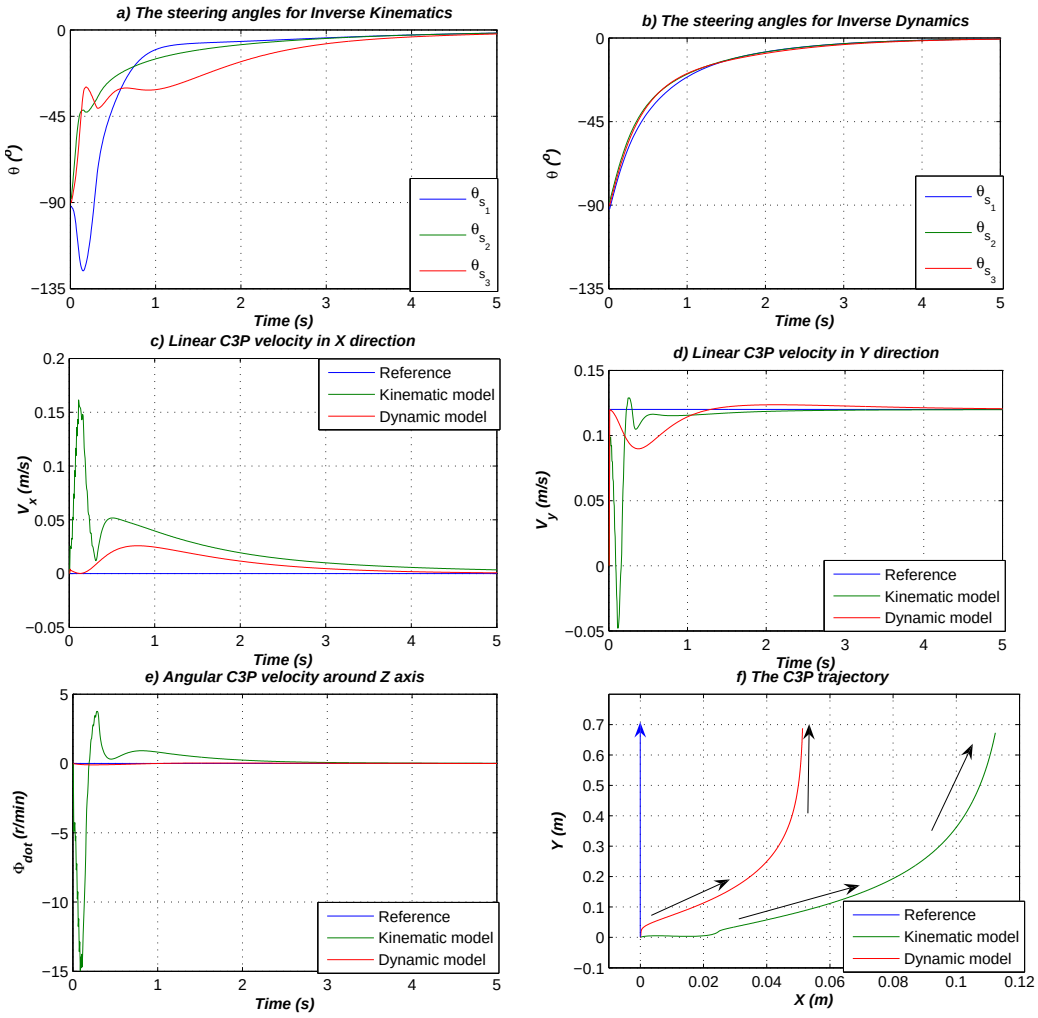

Figure 4.8. Comparing dynamic and kinematic inverse solutions for mobility in y direction from an initial singular condition

The 'Dynamics and Kinematics fusion' block has one main function, fusing between the inverse kinematic (2.10) and the inverse dynamic solution (4.10). In the case of using the inverse dynamic solution separately, the wheel torques found in the vector τ_D are not symmetrically accumulated due to the friction and slippage problem. These unsymmetrical accumulations results in unstable behaviors of the C3P. Therefore the inverse kinematic solution is used to stabilize the behavior and overcome the instabilities (Fig. 4.9). The fusion criterion depends mainly on the effect of each solution, for example the inverse dynamic solution main function is

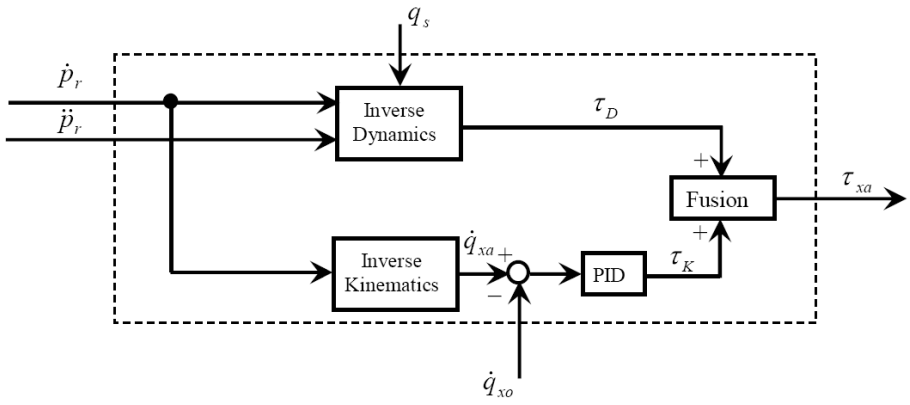

Figure 4.9. Dynamics and Kinematics fusion block

solving the singularity problem. Therefore it is needed in the transient phase in case of a singular condition. However, the inverse kinematic solution is sufficient in the steady state phase in the case of the non-singular condition. As a result, the fusion equation is a function of the singularity indicator Ψ, as described below

$$\tau_{xa} = (1 - cos(|\Psi|))^{\lambda_1}\tau_D + cos(|\Psi|)^{\lambda_2}\tau_K, \tag{4.16}$$

where τ_D is the wheel torque resulted from the inverse dynamic solution and τ_K is the torque resulting from the wheel velocity axes control. The fusion parameters λ_1 and λ_2 are tuned manually. The initial condition used in Figure (4.6) are used as well in the next example with the same input robot velocities. The parameters λ_1 and λ_2 have the same value, which is '0'. The output results in Figure(4.10) shows that the fusion between the inverse dynamics and the inverse kinematics solutions overcome the singularity problem. Where the initial steering angles value is $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = 0^o$ and the reference robot velocity $\dot{\mathbf{p}}_r = [0.12(m/s)\ \ 0(m/s)\ \ 0(r/min)]^T$ should result in steering for the wheel angles to reach the value of -90^o.

It is noticeable that the steering wheels are adjusted to the steady state value -90^o and the robot velocities follow their required value after almost two seconds. The based control system starts by using the inverse dynamics model because the singularity indicator $|\Psi|$ indicates that the robot lies in singular configuration. Then the system switches smoothly to the inverse kinematic solution after $0.75(s)$

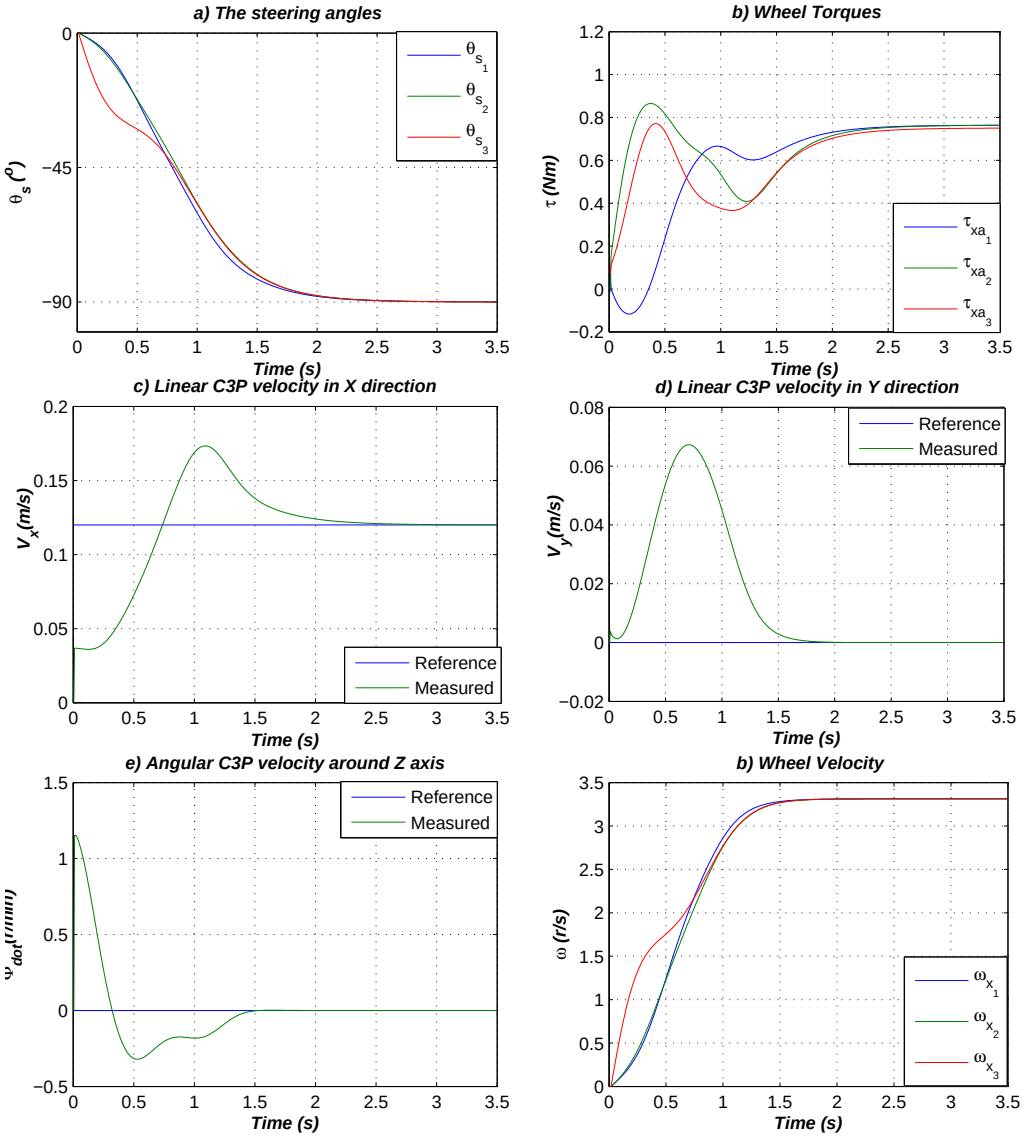

Figure 4.10. Simulation results using Dynamics and Kinematics fusion

with respect to the values of λ_1 and λ_2. The switching is noticed in steering angles Figure (4.10-a) and the wheel angular velocities Figure(4.10-f).

4.2.3 Position Controller

The position controller presented in this section is similar to the one in Section (3.5), where it has three main controlled variables: $\dot{x}$, $\dot{y}$ and $\dot{\phi}$. The po-

sition control problem can be stated as follows: given position goal coordinates vector $\mathbf{p}_g = \begin{bmatrix} x_g & y_g & \phi_g \end{bmatrix}^T$ and initial position reference coordinates vector $\mathbf{p}_i = \begin{bmatrix} x_i & y_i & \phi_i \end{bmatrix}^T$.

The robot is assumed to move in a straight line between the initial and the goal points. Therefore, there is another variable that should be controlled, which is δ_{er}. the variables $\delta_{er} = \vartheta - \delta_g$, where $\vartheta = \arctan(\frac{-y_g}{x_g})$ and $\delta_g = \arctan(\frac{-y_e}{x_e})$ (Fig.4.11)

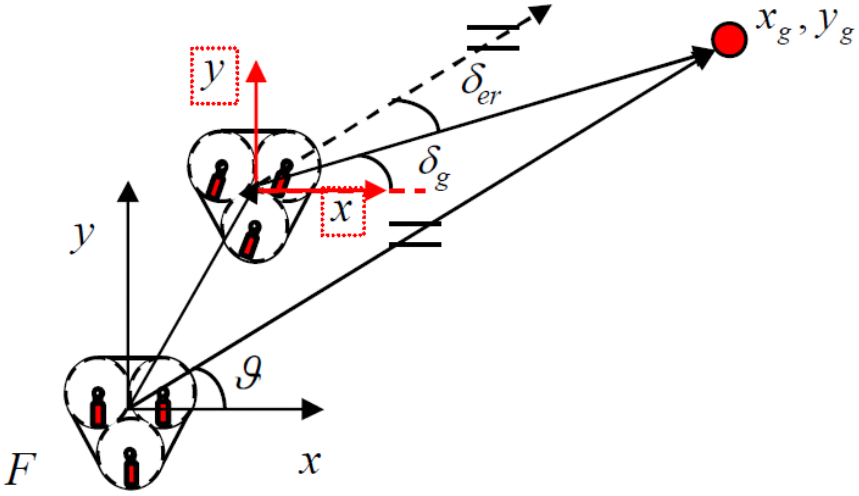

Figure 4.11. Robot Position representation.

From the previous section, the velocity controller succeeded in driving the robot in the floor coordinates but aimed for the wrong goal point with a distance error. The angle δ_{er} is the difference angle between the direction of the robot and the error vector, with reference to the mass point of the robot at every instant. The position controller should simply be the proportional of the distance error between the initial and the goal points in x direction (x_e) and y direction (y_e), and the error in the orientation angle ϕ (ϕ_e). The controller delivers robot reference velocities value to the velocity controller, which vanishes by the time the robot reaches the goal value. The position controller used in the practical implementation is exponentially a function of the robot displacement error, shown as the following

$$\dot{x} = K_x x_e (1 - e^{(-\|x_e\|)})^{\mu_x}, \tag{4.17}$$

$$\dot{y} = K_y y_e (1 - e^{(-\|y_e\|)})^{\mu_y}, \tag{4.18}$$

and

$$\dot{\phi} = K_\phi \phi_e + K_{er} \delta_{er} \tag{4.19}$$

where the angle δ_{er} is the difference angle between the direction of the robot and the error vector with reference to the mass point of the robot at every instant.The parameters μ_x and μ_y are tuned to increase the stability of the controller. The noticeable effect of the controller parameters tuning appears mainly in the robot trajectory. For example, Figure (4.12) is the simulation output for driving the C3P from initial coordinates of $\mathbf{p}_i = [0(m) \;\; 0(m) \;\; 0(^o)]^T$ to the goal point coordinates of $\mathbf{p}_g = [-4.5(m) \;\; -4.5(m) \;\; 90(^o)]^T$ and the initial steering angles are $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = 45^o$.

The control parameters are set to be as the following; $K_x = 0.1$, $K_y = 0.1$, $K_\phi = 0.4$, $K_{er} = 0.1$ and $\mu_x = 0$ and $\mu_y = 0$. The steering angles were adjusted in the first couple of seconds to maintain the zero orientation angle error Φ_e, as shown in Figure(4.12 a & e). After the rotation of the C3P 90^o it starts driving in a $-x, -y$ direction to reach the zero linear displacement error as shown in Figure (4.12d).

The periority of rotating the robot to the reference angular position ($\dot{\phi} = 90^o$) is noticed by the parameter's ratio 4:1 (K_ϕ:K_x or K_y). There are two main factors that affect the C3P trajectory: a) the position control parameters, and b) the initial steering angles values. The controller parameters K_ϕ and K_{er} are more effective on the robot trajectory over the linear parameters K_x and K_y. Where good adjustment for the parameters K_{er} will yield to a fast and stable position trajectory, bad adjustment to the parameter K_ϕ may yield to unexpected oscillations and loops in the robot trajectory. The effect of these two parameter adjustments is further demonstrated in Figure (4.13) where different values of the parameters are assigned.

First the same example with the same parameters is tested four times with different values for K_ϕ and K_{er}. The goal coordinates are $\mathbf{p}_g = [-3.5(m) \;\; -4(m) \;\; 90(^o)]^T$. Figure (4.13b) resulted in K_ϕ =0.2 and $K_{er} = 0.5$, where the high value of K_{er} tried to follow the reference trajectory (dashed arrow) as fast

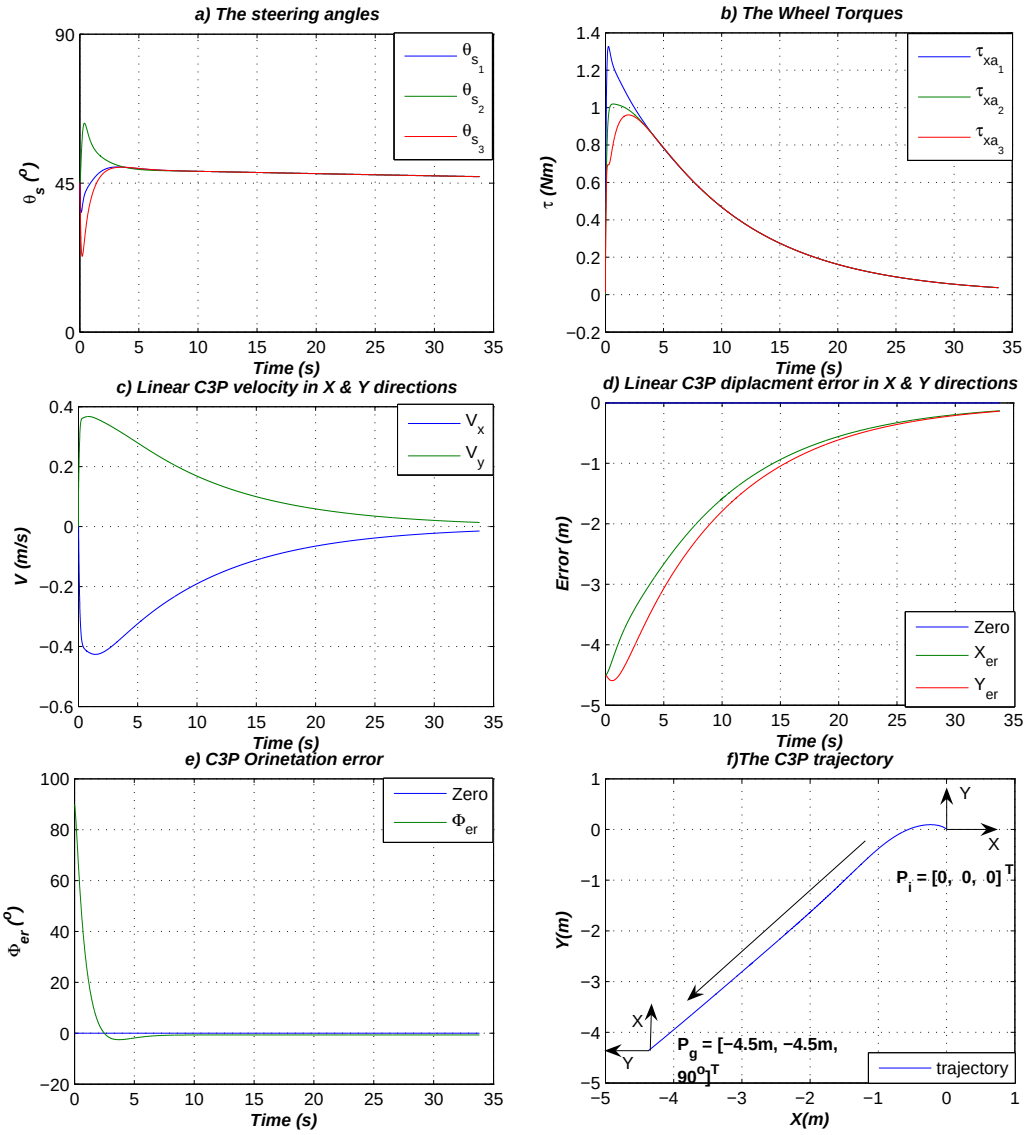

Figure 4.12. Position controller results for driving between $\mathbf{p}_i = [0m, 0m, 0^o]^T$ to $\mathbf{p}_g = [-4.5m, -4.5m, 90^o]^T$.

as possible. The robot ended in an almost 0.5m radius circle around the initial point before driving in a sinusoidal shape towards the goal point. The trajectory shown in Figure (4.13a) has the parametric values of K_ϕ =0.2 and K_{er} = 0.2. The reduction of K_{er} value succeeded in avoiding the C3P circle trajectory, yet the trajectory oscillates near the reference dashed trajectory. The K_{er} parameter

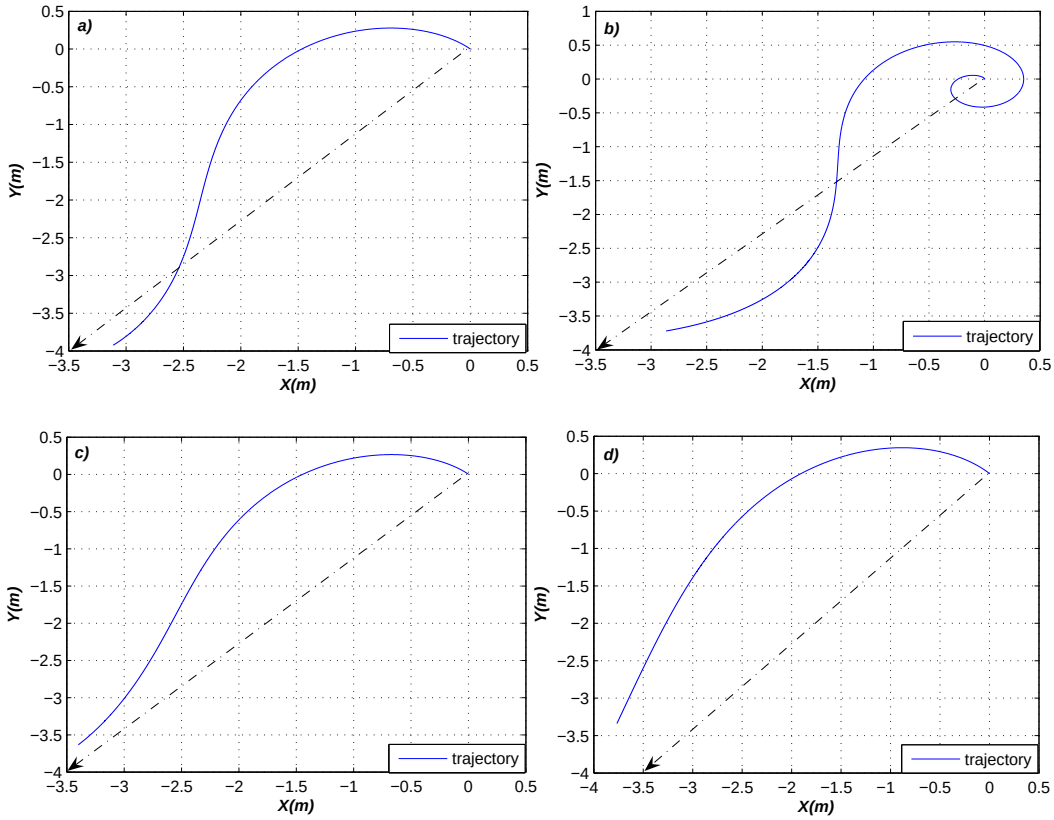

Figure 4.13. The effect of the parameters K_ϕ and K_{er} on the robot trajectory.

was reduced to the value of 0.1, which affected the trajectory as shown in Figure (4.13c) maintaining smooth motion towards the goal. However, if the parameters are adjusted to smaller values as in Figure (4.13d), where K_ϕ =0.1 and $K_{er} = 0.01$, the robot drives in an ellipse away from the goal coordinates.

The steering angles' initial values are the second factor affecting C3P trajectory. In the transient phase, the C3P needs space and time to adjust its steering angles from their initial values to the desired values. As a result, the behavior of the C3P trajectory is not the same for different initial steering angles values. Therefore, the first example is carried out for different initial angles with K_ϕ =0.4 and $K_{er} = 0.1$, and their simulated output trajectories are shown in Figure (4.14a). By zooming on the transient phase, the difference in trajectories will be more clear in Figure (4.14b), such differences are important to be dampened in case of driving in narrow

spaces.

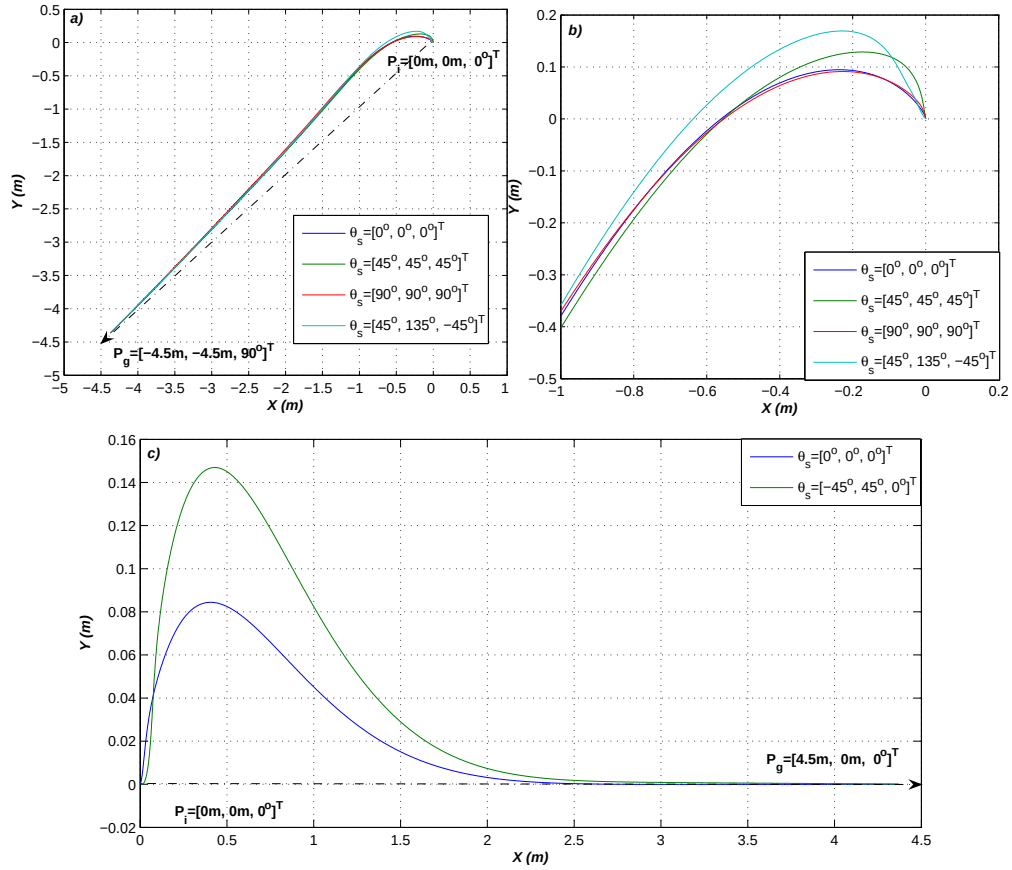

Figure 4.14. the effect of initial steering angles values on the C3P trajectories for two different examples.

Another example with the goal coordinates $\mathbf{p}_g = [4.5(m) \quad 0(m) \quad 90(^o)]^T$ and the same controller parameters is tested. Figure (4.14c) shows the trajectories generated from a singular condition $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = 0^o$ and a non-singular condition $\theta_{s_1} = -45^o$, $\theta_{s_2} = 45^o$ and $\theta_{s_3} = 0^o$. The non singular condition yields more displacement errors than the singular condition to reach the reference dashed trajectory. Such trajectories lead to the conclusion that that the non singular initial condition does not have to give better trajectories than the singular ones. The main conclusion is that the behavior of the C3P during the steering angles adjustment phase cannot be predicted due to the different initial steering values and its actuated velocities.

4.3 Summary

This chapter explained and demonstrated the derivation of the inverse dynamic solution for the C3P using the Euler-Lagrangian method. The solution was used in the velocity and position control loops in several simulation examples. The simulation outputs showed the effective performance of the solution with the controllers and also showed that the behavior of the C3P trajectory cannot be accurately predicted (during the steering wheel angles adjustment phase) due to the following factors: the initial steering angles values and the initial wheel and steering angular velocities values.

Chapter 5

The Lyapunov Stability Analysis

5.1 Introduction

The direct method of Lyapunov used in this chapter is the most general method for determining the stability of nonlinear systems. By this method the stability of the system can be determined without solving the state equations. This is quite advantageous because solving nonlinear state equations is usually very difficult. The main idea behind the direct method of Lyapunov is that the energy of a free mechanical system is always dissipated (due to friction, etc) and never restored. The system must eventually settle down to an equilibrium point [92],[93],[94] and [95].

We shall primarily consider what is often called 'Lyapunov's Direct Method'. It involves finding a 'Lyapunov function' for a system. If such a function exists, then the system is stable. A related result shows that if a similar function exists, it is possible to show that a system is unstable. The tricky part about Lyapunov's direct method is the search for an appropriate function.

5.2 The Lyapunov Function

Consider the C3P model described with the block signal flow presented in Fig.(5.1). It is a closed loop system between the measured robot velocities $\dot{\mathbf{p}}_m = [\dot{x}\ \dot{y}\ \dot{\phi}]^T$ and the input goal coordinates $\mathbf{p}_g = [x_g\ y_g\ \phi_g]^T$.

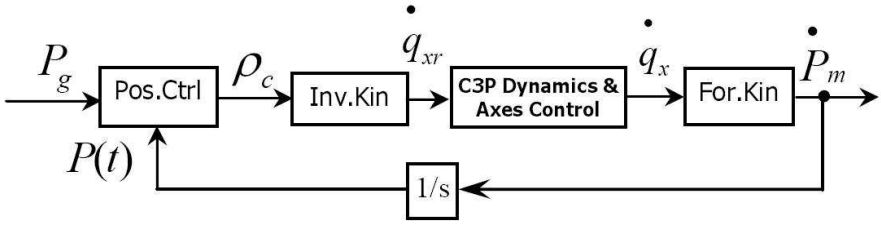

Figure 5.1. The C3P open control structure

where the block (Pos.Ctrl) is the robot position controller, as seen from Figure (5.1), the system contains the robot kinematics/dynamics models and the relation between $\dot{\mathbf{p}}_m$ and ρ_c can be deducted in three steps. First the Forward Kinematics, which was developed in Chapter (2) and represented by equation 2.15 as follows

$$\dot{\mathbf{p}}_m = J_f(\mathbf{q}_s)\ \dot{\mathbf{q}}_x, \tag{5.1}$$

The second step is a relation between the reference and output wheel angular velocities, which is described by the dynamic model mentioned in Chapter 2 through the following equation

$$\dot{\mathbf{q}}_x = M^{-1}(\mathbf{q}_s)\ \dot{\mathbf{q}}_{x_r}, \tag{5.2}$$

where the matrix $M(\mathbf{q}_s)$ is a non singular matrix containing the robot mass and inertia parameters in addition to the axes level control parameters. The third step is the inverse kinematic solution (2.10), which is described with the following relation corresponding to the input control signal ρ_c

$$\dot{\mathbf{q}}_{x_r} = J_{in_x}(\mathbf{q}_s)\ \rho_c. \tag{5.3}$$

From equations (5.1), (5.2,) and (5.3) the relation between the measured robot velocities and the control signal can be described as

$$\dot{\mathbf{p}}_m = J_m\ \rho_c \tag{5.4}$$

where the matrix $J_m(\mathbf{q}_s) = J_f(\mathbf{q}_s)\ M^{-1}(\mathbf{q}_s)\ J_{in_x}(\mathbf{q}_s)$ is 3×3 matrix containing the robot dynamics and geometrical mathematical representation. Such a model

is a complex non-linear model, its behavior depends on the control signal resulting from ρ_c, which is described as

$$\rho_c(t) = \begin{bmatrix} U_{cx} \\ U_{cy} \\ U_{c\phi} \end{bmatrix} = \begin{bmatrix} K_x(x_g - x(t))(1 - e^{(-\|x_g - x(t)\|)}) \\ K_y(y_g - y(t))(1 - e^{(-\|y_g - y(t)\|)}) \\ \left\{ \begin{array}{c} K_\phi(\phi_g - \phi(t))(1 - e^{(-\|(\phi_g - \phi(t))\|)}) \\ +K_{er}\left[tan^{-1}\left(\frac{-y_g}{x_g}\right) - tan^{-1}\left(\frac{-(y_g - y(t))}{(x_g - x(t))}\right)\right] \end{array} \right\} \end{bmatrix} \tag{5.5}$$

where K_x, K_Y, K_Φ and K_{er} are the position control parameters and in these analysis K_{er} will always be zero. The robot dynamics system discribed in (5.4) and (5.5) satisfies the following

$$\dot{\mathbf{p}}_m = F(\mathbf{p}_m(t)), \qquad \mathbf{p}_m(t) = \bar{\mathbf{p}}_o \tag{5.6}$$

where $\bar{\mathbf{p}}_o \in R^3$ is an equilibrium of (5.6) if $F(\bar{\mathbf{p}}_o) = 0$, i.e if $\mathbf{p}_m(t) = \bar{\mathbf{p}}_o$ is a solution of (5.6). In this system $\bar{\mathbf{p}}_o = \mathbf{p}_g$ is the stable equilibrium of (5.6) if for every $\epsilon_l > 0$ there sxists $\delta_l > 0$ such that all maximal solutions $\mathbf{p}_m = \mathbf{p}_m(t)$ of (5.6) with $|\mathbf{p}_m(0) - \bar{\mathbf{p}}_o| \leq \delta_l$ are defined for all $t \geq 0$, and satisfy $|\mathbf{p}_m(t) - \bar{\mathbf{p}}_o| < \epsilon_l$ and $\mathbf{p}_m = [x\ y\ \phi]^T$.

The energy equation is used and recommended as a Lyapunov function candidate by several mobile robot field researchers [95], because it is a square function yielding to a zero value if the robot reaches its goal coordinates. Therefore the energy equation is used under the assumption that the steering angles have the same values, which represent the steady state phase

$$E = \frac{1}{2}\, m_p \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}^T \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \dot{\phi} \end{bmatrix}^T I_p \begin{bmatrix} \dot{\phi} \end{bmatrix}, \tag{5.7}$$

where $[\dot{x}\ \dot{y}]^T$ and $[\dot{\phi}]$ are extracted from equation (5.4) with $\dot{\mathbf{p}}_m = [\dot{x}\ \dot{y}\ \dot{\phi}]^T$. By using the aid of Maple, the energy equation or the candidate Lyapunov equation

will be written in the follwoing form

$$V(t) = \rho_c^T(t)\ A_v(\mathbf{q}_s)\ \rho_c(t) \tag{5.8}$$

where

$$A_v(\mathbf{q}_s) = \begin{bmatrix} a_{xx} & a_{xy}/2 & b_{x\phi}/2 \\ a_{xy}/2 & a_{yy} & c_{x\phi}/2 \\ b_{x\phi}/2 & c_{x\phi}/2 & a_{\phi\phi} \end{bmatrix}. \tag{5.9}$$

Each matrix element is a function of the wheels steering angles $\mathbf{q}_s$ and represented in Appendix (D). As shown, equation (5.8) is a quadratic equation which is always positive definite, hence

$$V(x, y, \phi, t) \geq 0 \tag{5.10}$$

for all possible values of x, y, ϕ and $\mathbf{q}_s$. Thus, it is proven that the chosen function is a Lyapunov candidate. The next step proves that the function is negative definite or negative semi-definite. The Lyapunov function's first derivative is the main factor for such a test

$$\dot{V}(x, y, \phi, t) = 2\ \rho_c^T\ A_v(\mathbf{q}_s)\ \dot{\rho}_c + \rho_c^T(t)\ \dot{A}_v(\mathbf{q}_s, \dot{\mathbf{q}}_s)\ \rho_c \tag{5.11}$$

In the sense of Lyapunov, if the function $V(x, y, \phi, t)$ is locally positive definite and its first derivative $\dot{V}(x, y, \phi, t) \leq 0$ locally in (x, y, ϕ) and for all t, then the origin of the system is locally stable. From equation (5.5) it is apparent that

$$\dot{V} = 0 \quad if\ \mathbf{p}_m = \bar{\mathbf{p}}_o = \mathbf{p}_g \tag{5.12}$$

for all possible values of $\mathbf{p}_m$, $\mathbf{q}_s$ for all values of t. Condition (5.12) shows that the Lyapunov function could be asymptotically stable if its derivative $\dot{V}$ is negative definite everywhere except at $\mathbf{p}_m = \mathbf{p}_g$. For the function to be Lyapunov candidate, the inquality condition $\dot{V}(x, y, \phi, t) < 0$ must be true. To illustrate such conditions the candidate function will be modified. First, the control signals described in the vector $\dot{\rho}_c$ found in the function's first term should be broken in the following equation ($\dot{\rho}_c = B_v\ \dot{\mathbf{p}}_m$). The matrix $B_v(t)$ contains the instantaneous position coordinates and the control parameters. The second step is to separate the steering

angular velocity $\dot{\mathbf{q}}_s$ from $\dot{A}_v(\mathbf{q}_s, \dot{\mathbf{q}}_s)$ matrix, as shown

$$\dot{V}(x, y, \phi, t) = 2\ \rho_c^T(t)\ A_{vb}(\mathbf{q}_s)\ \dot{\mathbf{p}}_m + \rho_c^T(t)\ A_{vd}\ \dot{\mathbf{q}}_{sm}\ \rho_c(t) \tag{5.13}$$

where $A_{vb}(\mathbf{q}_s) = A_v(\mathbf{q}_s)\ B_v$. The matrix $\dot{A}_v$ is decomposed into two matrices $\dot{A}_v(\mathbf{q}_s, \dot{\mathbf{q}}_s) = A_{vd}\ \dot{q}_{sm}$, where the $\dot{q}_{sm}$ contains a set of steering angular velocities $\dot{\mathbf{q}}_s$ vectors

$$\dot{q}_{sm} = \begin{bmatrix} \dot{\mathbf{q}}_s & \dot{\mathbf{q}}_s & \dot{\mathbf{q}}_s \\ \dot{\mathbf{q}}_s & \dot{\mathbf{q}}_s & \dot{\mathbf{q}}_s \\ \dot{\mathbf{q}}_s & \dot{\mathbf{q}}_s & \dot{\mathbf{q}}_s \end{bmatrix}. \tag{5.14}$$

The C3P motion behavior consists of two phases: a) steady state where the steering angles yield the same constant value, b) non-steady state (transient phase or during steering angles disturbances), where the angles are not equal and and have a variable value.

For the steady state phase the steering angular velocity vector $\dot{\mathbf{q}}_s$ will be zero, due to the steering angles constant value. For the system to be stable, the following condition must be true

$$\dot{V}(x, y, \phi, t) = 2\ \rho_c^T(t)\ A_{vb}(\mathbf{q}_s)\ \dot{\mathbf{p}}_m\ \leq 0 \quad \dot{\mathbf{q}}_s = [0\ 0\ 0]^T \tag{5.15}$$

If the steering angular velocities have a non-zero value then the condition

$$\dot{\mathbf{p}}_m\ \leq -\tfrac{1}{2}\ A_{vb}^{-1}(\mathbf{q}_s)\ \dot{A}_v(\mathbf{q}_s, \dot{\mathbf{q}}_s)\ \rho_c(t) \quad \dot{\mathbf{q}}_s \neq [0\ 0\ 0]^T \tag{5.16}$$

is necessary for the system be stable. In this thesis, the system stability will be discussed under the first condition represented in equation (5.15), where the steering angles are assumed to be equal. Such a condition is considered because it is important for the system to be stable when an initial singularity condition exists, where the steering angles yield the same value and the steering angular velocities are zero. Furthermore, it is important for the system to be stable during the motion towards the goal point when all the angles are nearly equal and the steering angular velocities $\dot{\mathbf{q}}_s$ are assumed to have a very small value or almost zero. The second condition in equation (5.16) will be investigated further in future work.

5.3 Numerical Analysis

The steering angles are not controlled directly by the position controller. These angles values operate the system model at different operating points since the system is non-linear.

Since the steering angles are assumed to be equal, the steering angular velocities $\dot{\mathbf{q}}_s$ are assumed to have a very small value or almost zero and the function will be tested for all possible angle values. The effect of the steering angular velocities $\dot{\mathbf{q}}_s$ appears during the disturbances phase or when the C3P changes it direction of motion. An operation of this kind will be discussed in future researches.

Figure (5.2) shows that the values of the $\dot{V}$ are always negative for all possible values of θ_s. For a such representation the state variables and their derivatives are

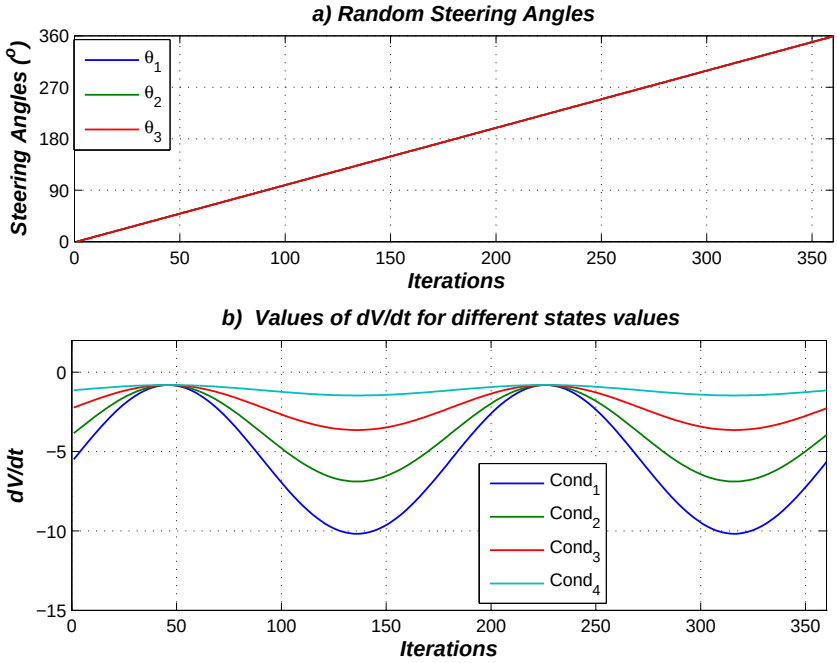

Figure 5.2. The values of $\dot{V}$ for equal uniform values of steering angles

considered for several conditions. The conditions changes the states of the $x(t)$ and $y(t)$, while the rest are $\phi(t) = \pi/5\ rad$, $\dot{x}(t) = 0.5\ m/s$, $\dot{y}(t) = 0.5\ m/s$, $\dot{\phi}(t) = 0.2\ r/min$, $K_x = 0.5$, $K_y = 0.5$ and $K_\Phi = 0.2$. The different position values are

$cond_1$: $x(t) = 0.5\ m$ and $y(t) = 0.5\ m$

$cond_2$: $x(t) = 1\ m$ and $y(t) = 1\ m$

$cond_3$: $x(t) = 1.5\ m$ and $y(t) = 1.5\ m$

$cond_4$: $x(t) = 2\ m$ and $y(t) = 2\ m$

From the figure, the following is concluded: the Lyapunov function exists for all steering angle values and different state conditions ($\dot{V}(t) \leq 0$), which is negative semi-definite. The main questions is: what if the steering angles are not equal? For example, if external disturbances are applied to the system during the steady state condition, the steering angles values will be influenced and different values will result.

The function represented in equation (5.7) may not be a Lyapunov function if the steering angles have different values. Since the condition $V > 0$ is always true, the same function is used in the stability determination. However, if $\dot{V} > 0$ is true, then the system will be considered unstable, if $\dot{V} < 0$, then the system is stable.

First of all the system states are considered constant with the values of $x(t) = 1\ m,\ y(t) = 1\ m,\ \phi(t) = \pi/5\ rad,\ \dot{x}(t) = 0.5\ m/s,\ \dot{y}(t) = 0.5\ m/s$, and $\dot{\phi}(t) = 0.2\ r/min$. The first example is to test the system with uniform steering angles values, but shifted from each other with 90^o, as shown in Figure(5.3-a). On the other hand, Figure(5.3-b) represents the positive values of $\dot{V}$ if they exist. It is quite noticeable that the function has positive values for some steering angles values combination.

These combinations are changed if at least one of the position control parameters (K_x, K_y,K_Φ) changes its value as shown in Figure(5.3-b). The parameter K_x value is set to five different values, while the other two parameters are kept unchanged. The three steering angles combinations are changed and presented in Table5.1. It is concluded from the table that as the value of K_x increases, the set of steering angles values combinations elements decreases. For example, when $K_x = 0.3$, the values set for each steering angle contained four different values ($\theta_{s_1} = 16^o - 20^o$,$\theta_{s_2} = 106^o - 110^o$ and $\theta_{s_3} = 196^o - 200^o$), while at $K_x = 0.7$ there were only two values for each steering angle($\theta_{s_1} = 4^o - 5^o$,$\theta_{s_2} = 94^o - 95^o$ and $\theta_{s_3} = 184^o - 185^o$).

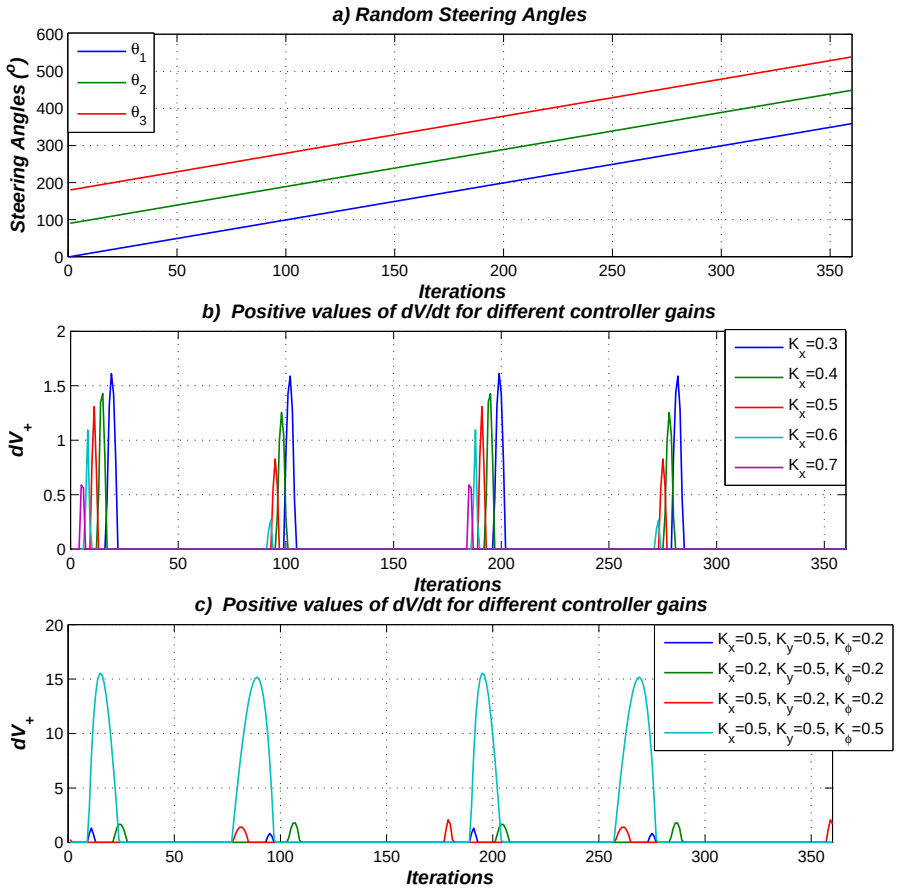

Figure 5.3. The values of $\dot{V}$ for non-equal uniform values of steering angles

Throughout the next examples the stability space will be searched for random values of θ_{s_1} and θ_{s_3}. The steering angle θ_{s_2} value shows where and how it affects the existence of the Lyapunov space (system stability) as shown in Figure (5.4-a). The figure is a 3D space for 20,000 random values of θ_{s_1} and θ_{s_3}. The Z-axis is the $\dot{V}$ discreet value; if $\dot{V}$ is smaller than zero, then the value is -1 and presented in blue '*'. If $\dot{V}$ is larger than zero, then the value is 1 and presented in red '*'. In the presented example the steering angle θ_{s_2} is chosen to be 205^o. As shown, $\dot{V}$ is positive definite for some combinations of θ_{s_1} and θ_{s_3}, which is considered as a possibly unstable region. These possibly unstable regions change with the variation of θ_{s_2} value. Figure (5.4(b-c-d)) shows how the instability regions will change in the Lyapunov space by the change of θ_{s_2}=205^o, 160^o and 250^o respectively. It is

Table 5.1. The steering angles values combination

K_x	$\theta^o_{s_1}$	$\theta^o_{s_2}$	$\theta^o_{s_3}$	K_x	$\theta^o_{s_1}$	$\theta^o_{s_2}$	$\theta^o_{s_3}$
0.3	16-20	106-110	196-200	0.4	12-15	102-105	192-195
0.3	99-103	189-190	279-283	0.4	95-99	185-189	275-279
0.3	96-200	286-290	376-380	0.4	192-195	282-285	372-375
				0.4	275-279	365-369	455-459
0.5	9-11	99-101	189-191	0.6	6-8	96-98	186-188
0.5	93-95	183-185	273-275	0.6	91-92	181-182	271-272
0.5	189-191	279-281	369-371	0.6	186-188	276-278	366-368
0.5	273-275	363-365	453-455	0.6	271-272	361-362	451-452
0.7	4-5	94-95	184-185				
0.7	184-185	274-275	364-365				

apparent that the unstable regions are different in shape, area, and position with respect to the three values of θ_{s_2}.

The values of the controller parameters also affect the position and the areas of the instability regions, which are depicted in Figure (5.5). By increasing K_x the instability (red'*') region can be reduced, as shown in Figure(5.5-d). But such change will differ from one steering angle value to another, or from one robot velocity to another, since $\dot{V}$ is a function of $x(t)$, $y(t)$,$\phi(t)$,$\dot{x}$,$\dot{y}$ and $\dot{\phi}$. These state values change instantaneously during every practical experiment for different initial conditions. Their values cannot be directly changed. On the other hand the control parameters can be reconfigured to change the unstable region position and area.

During each experiment, disturbances may affect the castor wheels, resulting in changing their steering angles. Such disturbances may occur due to sudden external force affecting the robot platform. The practical stability tests are presented in chapter (6).

5.4 Conclusion

The C3P kinematics/dynamics model is a complex nonlinear system in comparison to other existing wheeled mobile robots. It is affected by many disturbances like friction, slippage, or actuation conflict. Such disturbances result in changing the steering angles configuration. For studying the system stability, a Lyapunov direct method is considered. The candidate Lyapunov function presented in this work is

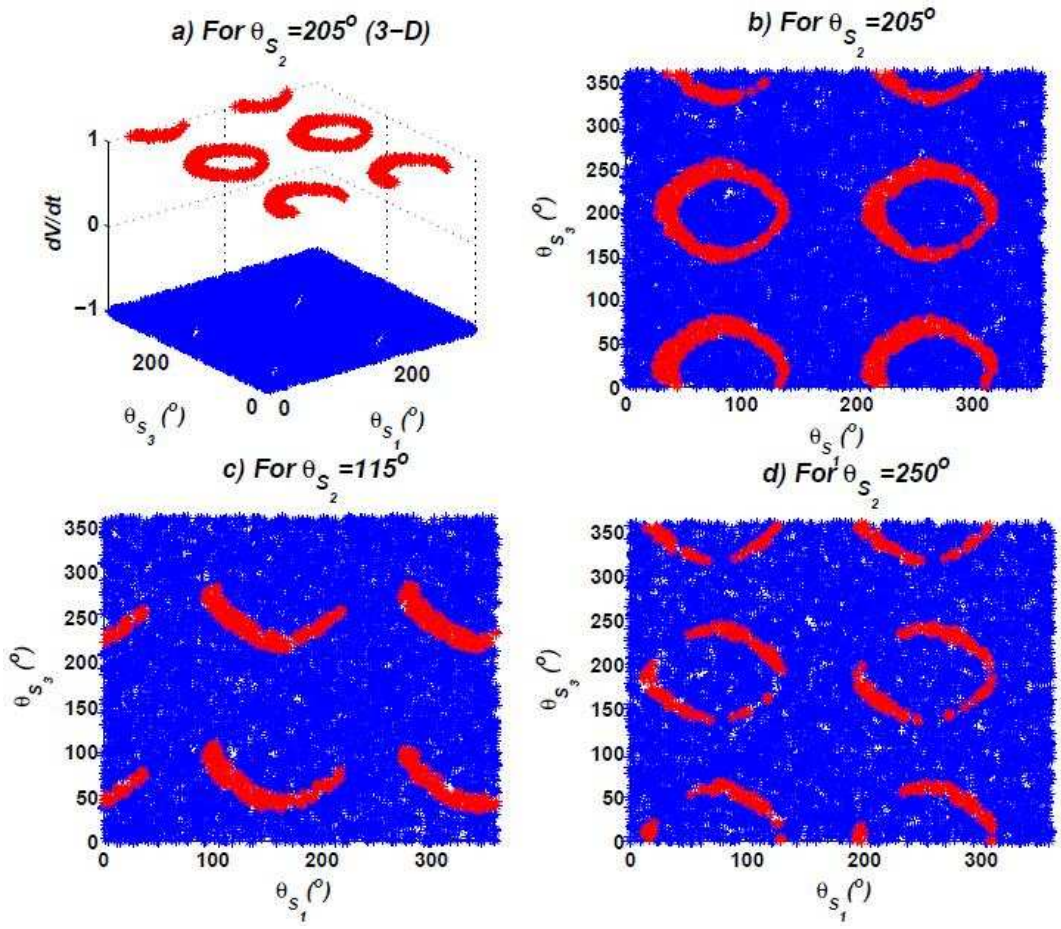

Figure 5.4. 3-D space $\dot{V}$ representation for random values θ_{s_1} and θ_{s_3}, unstable: *, stable: *

a quadratic one extracted from the robot kinetic energy equation.

The results discussed in this chapter show that the used Lyapunov function leads to two different regions, where the first derivative can be negative semi-definite or positive semi-definite. If the first derivative is negative semi-definite, a Lyapunov function exists and the system is always stable. On the other hand, if the first derivative is positive semi-definite, no statment on stability can be made using the candidated function. Therefore, this function cannot be considered as a Lyapunov function because its first derivative has positive values for some steering angles configurations.

For better performance the non-defined stability regions should be avoided, for

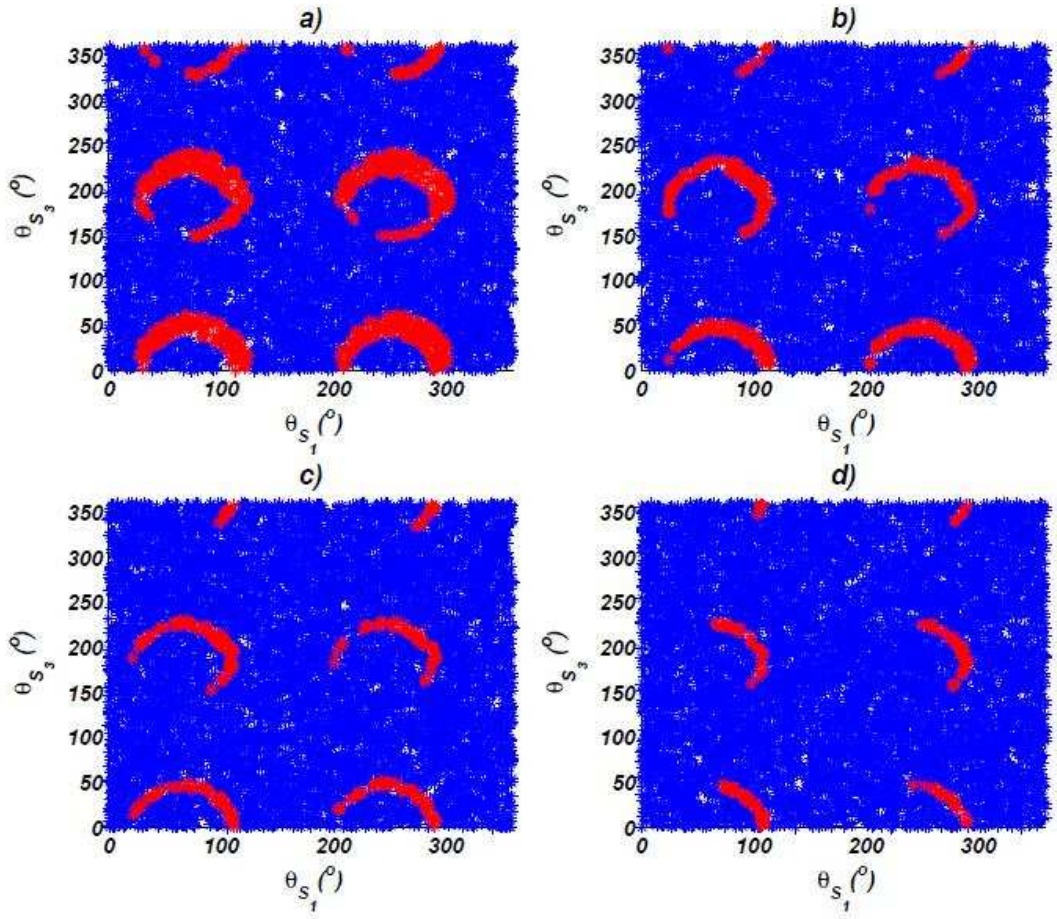

Figure 5.5. 2-D Lyapunov space with random θ_{s_1} and θ_{s_3} values for different K_x values at $\theta_{s_2} = 205^o$ values, a)K_x=0.5, b) K_x=0.9, c)K_x=1.1, d)K_x=1.5, unstable: *, stable: *

example, by predesigned trajectories that keep the system in the region where the Lyapunov function always exists.

Chapter 6

Implementation and Practical Results

6.1 Platform Hardware Configuration

The C3P platform is implemented in the Automation Laboratory. The practical platform used in this work is the first prototype of the C3P configuration shown in Figure(6.1).

Figure 6.1. The C3P practical prototype

The platform has three caster wheel units and each unit contains four main

components: a) Brush-less DC motor for the wheel angular velocity actuation, b) incremental encoders for sensing the angular velocity of the wheels, c) absolute encoders for measuring the wheel steering angle, and d) slip rings for signal transfer between the motors and the control cards. The caster wheel unit has two levels: the upper level (attached to the platform) and the lower level (attached to the wheel). The lower level is shown in Figure(6.2) with the DC servomotors, which are mounted on the driven axis of each wheel along with the incremental encoders.

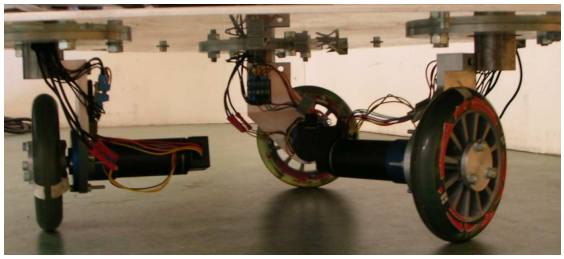

Figure 6.2. The lower level of the caster wheels units

The wiring of the each motor and each encoder is connected to a slip ring mounted on the upper level of the caster wheel unit shown in Figure (6.3).

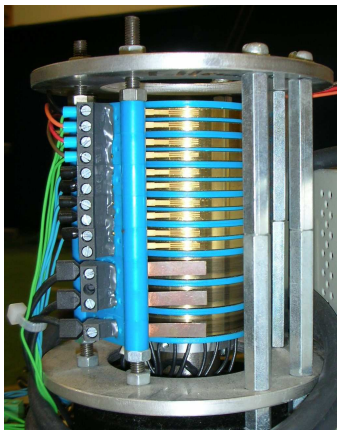

Figure 6.3. The slip rings

The slip rings have 12 rings with 16 Ampere and below the slip ring the absolute encoder is mounted (Fig. 6.4).

The motors are controlled by velocity control cards (Fig.6.5) mounted on the platform body. The controller is 4-Quadrant PWM Servo Amplifier, 24 V power

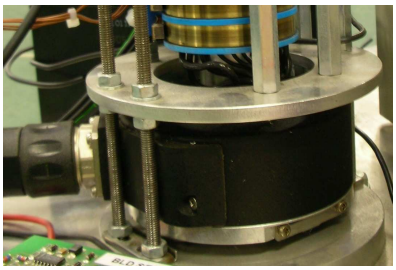

Figure 6.4. The absolute encoder

supply with rated speed 40,000 r/min.

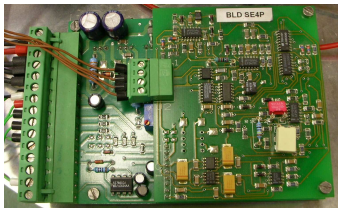

Figure 6.5. Velocity control cards

The sensed and control signals are transferred to an industrial computer 1 GHz speed running under QNX operating system.

6.2 Kinematics Based Controller Experiments

The simulation results for the kinematic solution presented in Chapter (3) illustrated the C3P performance and behavior. This section presents the C3P performance on a practical level, where the inverse kinematics solution along with its controllers are implemented on the C3P prototype. The control structure is presented in Figure (6.6), where the axes level control is the hardware motors' velocity cards and the WCE block is the Wheel Coupling Approach, and the Vel.Crtl block is the robot level velocity controller presented in Chapter (3)

The platform parameters are close to the simulation parameters and are presented in Table 6.1, which will be used in the experiments presented throughout this chapter. The first experiment is the testing of the coupling approach open loop structure (Fig. 6.6) with no Vel.Ctrl. The desired C3P velocity vector

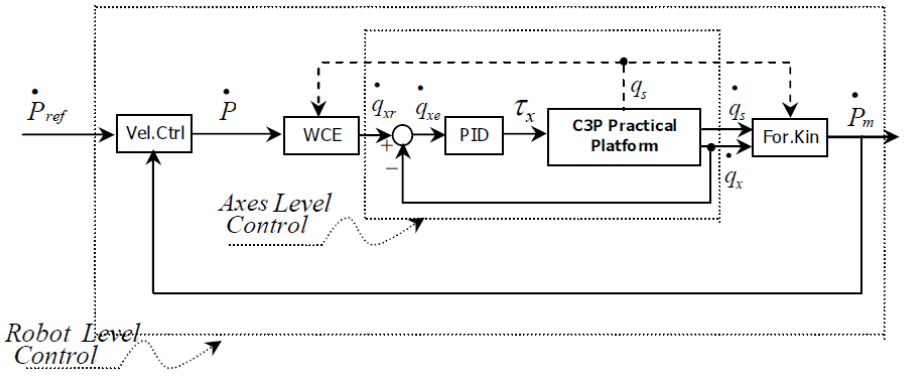

Figure 6.6. Kinematics Based Control Structure

$\dot{\mathbf{p}} = [0.12m/s\ 0m/s\ 0r/min]^T$. The initial steering angles values are then initial steering angles $\theta_{s_1} = \theta_{s_2} = -4^o$, $\theta_{s_3} = -7^o$, as shown in Figure(6.7-a). Such input yields the steering angles to reach -90^o (Fig.6.7-b). Figure (6.8) shows the practical results of the C3P prototype.

Table 6.1. The C3P parameters

C3P Parameters	Value	Units
h	0.343	m
d	0.04	m
r	0.04	m

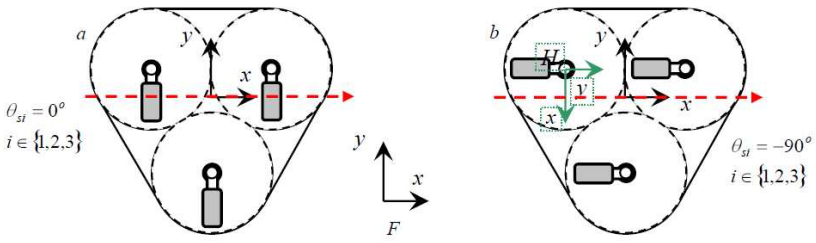

Figure 6.7. The Steering angles orientation for driving in X direction from configuration (a) to configuration (b)

The figure shows the steering angles trajectories reach the desired value -90^o (Fig. 6.8a). Some oscillations appear in each steering angles plot, as a result

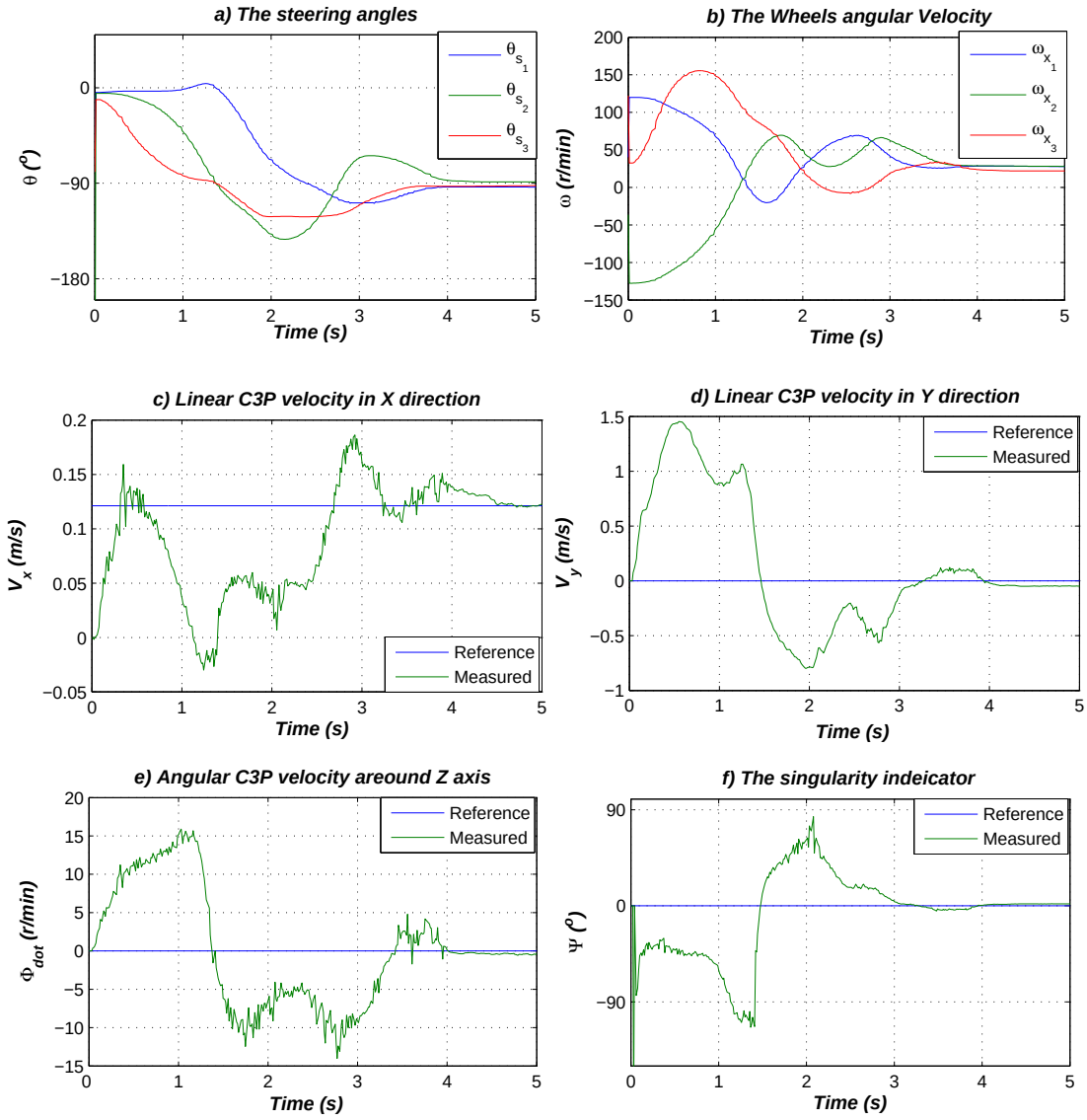

Figure 6.8. C3P practical results for driving in x direction with open loop WCE

oscillations in the wheels angular velocities occur (Fig. 6.8b). These oscillations do not exist in the simulation results for two main reasons: a) the friction and slippage are not included in the C3P dynamic model, and b) the considered assumptions in the WCE. The angular velocities of wheel one and two (ω_{x_1} & ω_{x_2}) or ($\dot{\theta}_{x_1}$ & $\dot{\theta}_{x_2}$) started with the same values but in the opposite direction to initiate coupling action for the steering of the steering angular velocity of wheel number three (ω_{s_3}

or $\dot{\theta}_{s_3}$).

In Figure 6.8c & d & e), it is normal for the velocity oscillations to appear during the steering angles adjustment phase, since there is no direct actuation on the wheels steering axes. The C3P velocity in the y direction has a noticeable steady state error that can be reduced using the robot velocity control loop. The steady state error also appears in the C3P angular velocity. The singularity indicator in Figure (6.8f) shows the singularity condition at the initial state with value of -90^o degrees, and then the value oscillates with the steering angle oscillations to reach 0^o.

The velocity control loop shown in Fig.(3.10)is used in the second experiment, where the robot reference velocities are $\dot{\mathbf{p}}_{ref} = [0.18m/s\ 0m/s\ 0r/min]^T$ as a desired velocity . Figure (6.9) presents the output results of the C3P implementation with the following robot velocity controller parameters $\lambda = 5$, $\epsilon = 8^o$, $K_{P_x} = 0.5$, $K_{P_y} = 0.5$, $K_{P_\phi} = 1.5$ and $K_{D_\phi} = 0$.

As mentioned in Chapter 3, the C3P velocity controller must achieve three main objectives: a) adjust the steering angles to the desired value, b) adjust the robot coordinates to match the floor coordinates without influencing the steering angles, c) the output robot velocity follows the input signals with respect to the floor frame of coordinates.

The steering angles reach the desired value in two seconds (Fig. 6.9a) but it started deviating again around the value of -90^o as a result of switching to the PID controller. The switching between the WCE regulator and the PID velocity controller (Chapter 3) is noticed in Figures (6.9c, d, e & f). When the singularity indicator indicates that the robot is out of singularity condition, the PID controller is initiated and the WCE is disabled. This appears in the angular robot velocity (6.9e), where its control signal is the main factor in adjusting the robot velocities in x and y directions. After one second the control signal is initiated and it affects the robot velocities performance (6.9c& d). The velocity in y direction with respect to the floor frame of coordinates is adjusted to follow the reference value with keeping the steering angles around their desired value. The same action appears in the output results in the x direction.

The third experiment is driving the C3P in y direction from initial steering angles of $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = -90^o$ (Fig. 6.11-a) and reference robot velocity vector

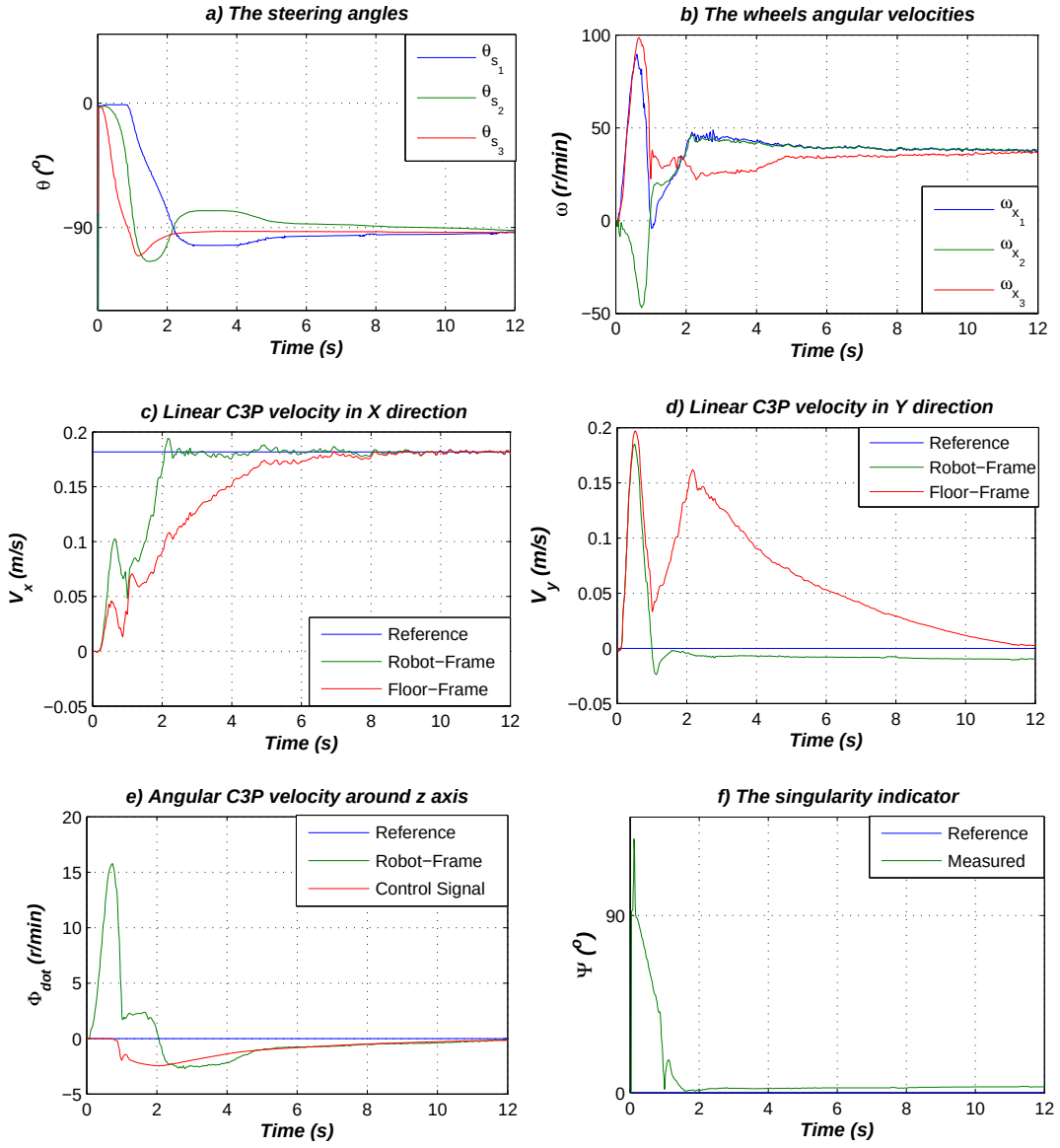

Figure 6.9. C3P results for driving in x direction with closed loop velocity control

$\dot{\mathbf{p}} = [0m/s\ -0.095m/s\ 0r/min]^T$ with the following controller parameters: $\lambda = 4^o$, $\epsilon = 5^o$, $K_{P_x} = 0.5$, $K_{P_y} = 0.5$, $K_{P_\phi} = 1.5$ and $K_{D_\phi} = 0$

The WCE regulator virtually actuates the steering angles to reach the configuration described in Figure(6.11-b). The steering angles trajectories are plotted in Figure(6.10a). From Figure(6.10b, c & d) it can be seen that the robot-frame

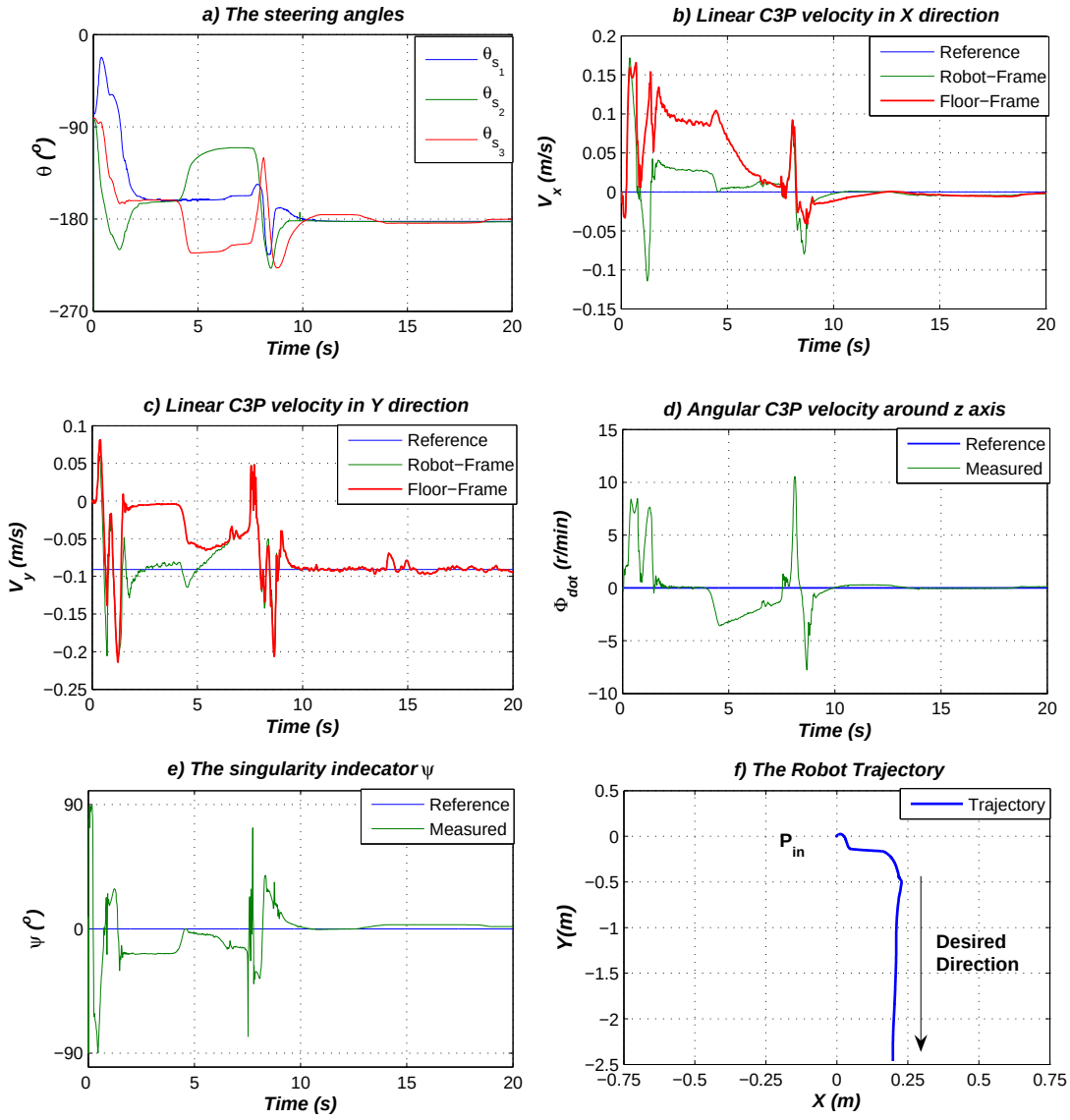

Figure 6.10. C3P results for driving in y direction

velocities are not aligned to the floor-frame velocities. At this moment the PID controller is activated and succeeds in settling the velocities responses with respect to the Floor.

The C3P trajectory shown in Figure(6.10f) has displacement errors in the x direction due to the wheel adjustment phase; however, the robot moves in the desired direction. It is apparent that the oscillations are produced in the steering

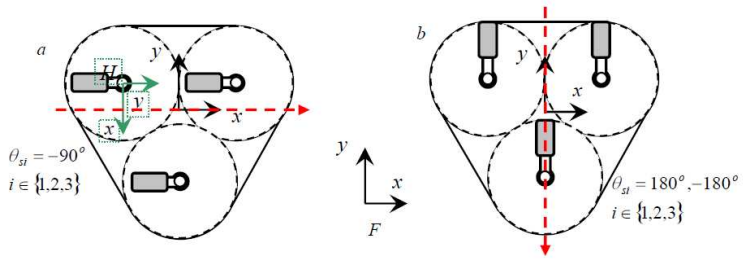

Figure 6.11. The Steering angles orientation for driving in -Y direction from configuration (a) to configuration (b)

angles performance due to the PID controller action, but the controller succeeds in dampening the oscillations and reaching the desired value -180^o.

The fourth experiment illustrates the performance of the position controller. The initial starting point had the coordinates of $\mathbf{p}_i = [0m\ 0m\ 0^o]^T$ and the goal point $\mathbf{p}_g = [-3m\ \ -3m\ 0^o]^T$, with initial steering angles of value around -50^o and control parameters of $K_x = 0.1$, $K_y = 0.1$, $K_\phi = 0.3$ and $K_{er} = 0.6$. The goal coordinates require the robot to drive in the $(-x, -y)$ direction and therefore the steering angles are adjusted to the reference value of -225^o, as shown in Figure(6.12d). The robot velocities in x and y directions started smoothly from zero value due to an exponential function being added to the controller for a smooth start (Fig.6.12b & c). The steering angles actuation phase generated oscillations in the robot velocities which affected the robot trajectory in Figure(6.12a) which is highlighted by the red circle. The errors in trajectory were compensated by the controller and the robot reached the goal point in 30 sec.

Figure 6.13 presents the C3P trajectories for two different experiments. The first experiment moves from the initial coordinates $\mathbf{p}_i = [0m\ 0m\ 0^o]^T$ to the goal co-ordinates $\mathbf{p}_g = [0m\ \ -3m\ 0^o]^T$, with random initial steering angles values ($\theta_{s_1} = 360^0$, $\theta_{s_2} = 25^o$, and $\theta_{s_3} = 270^o$). Such goal coordinates adjusted the steering angles to the desired value -180^o, as shown in Figure (6.13b(i)). The trajectory errors observed in Figure (6.13a(i)), that are highlighted with the red dotted circle, are the effect of the steering angles adjustment phase.

The second experiment is driving to the goal coordinates $\mathbf{p}_g = [4.2m\ 0m\ 0^o]^T$ from initial coordinates $\mathbf{p}_i = [0m\ 0m\ 0^o]^T$ from singular condition described by the initial steering angles of $\theta_{s_1} = \theta_{s_2} = \theta_{s_3} = 0^o$ (Fig. 6.13). The steering angles

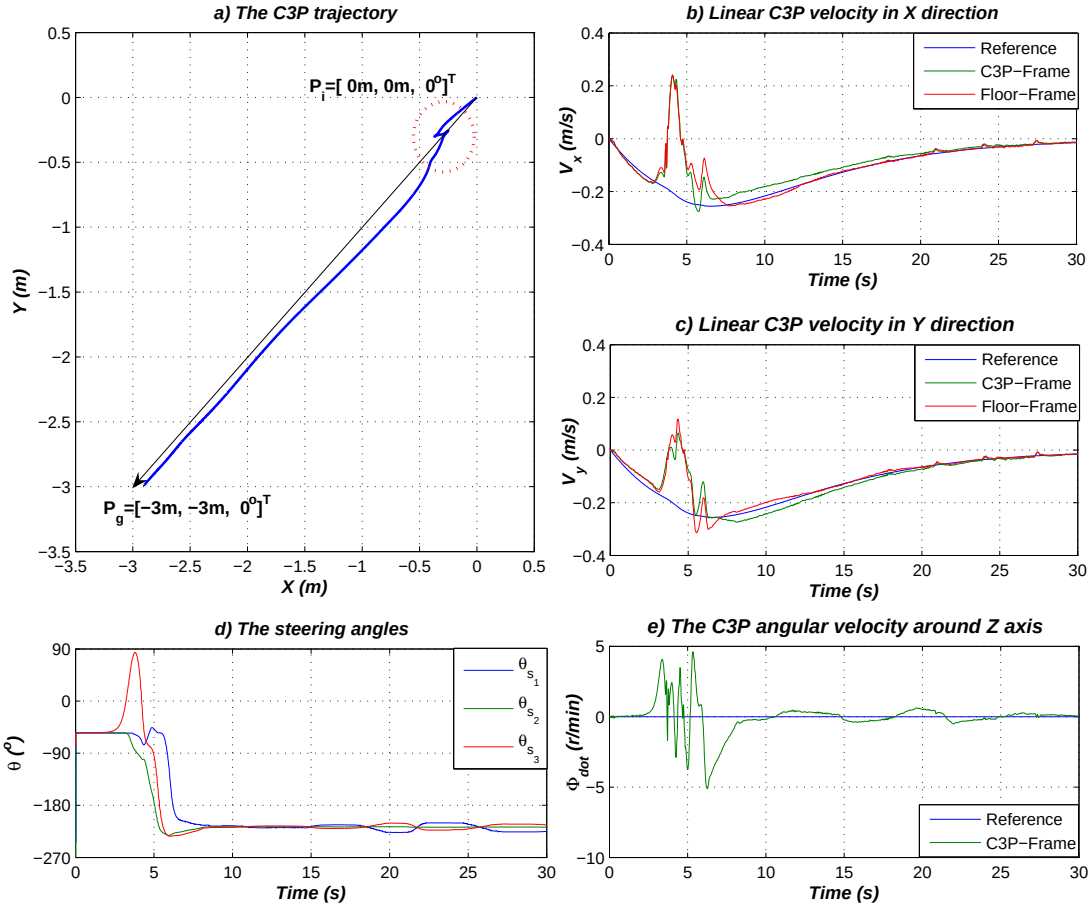

Figure 6.12. Position control experiment, driving from $\mathbf{p}_i = [0m\ 0m\ 0^o]^T$ to $\mathbf{p}_g = [-3m\ -3m\ 0^o]^T$

are adjusted to the desired value -90^o after some oscillations around it. These oscillations appear due to the action of the position controller to drive the robot to the goal coordinates, in addition to maintaining the zero robot orientation degree. The robot trajectory shown in Figure (6.13a(ii)) consumes time and displacement errors in adjusting the robot steering angles (red circle). However, the position control actions overcome these errors as far as possible and the robot reaches its goal. The main problem with the C3P is that the steering angles are the main factors affecting its mobility behavior and they cannot be predicted.

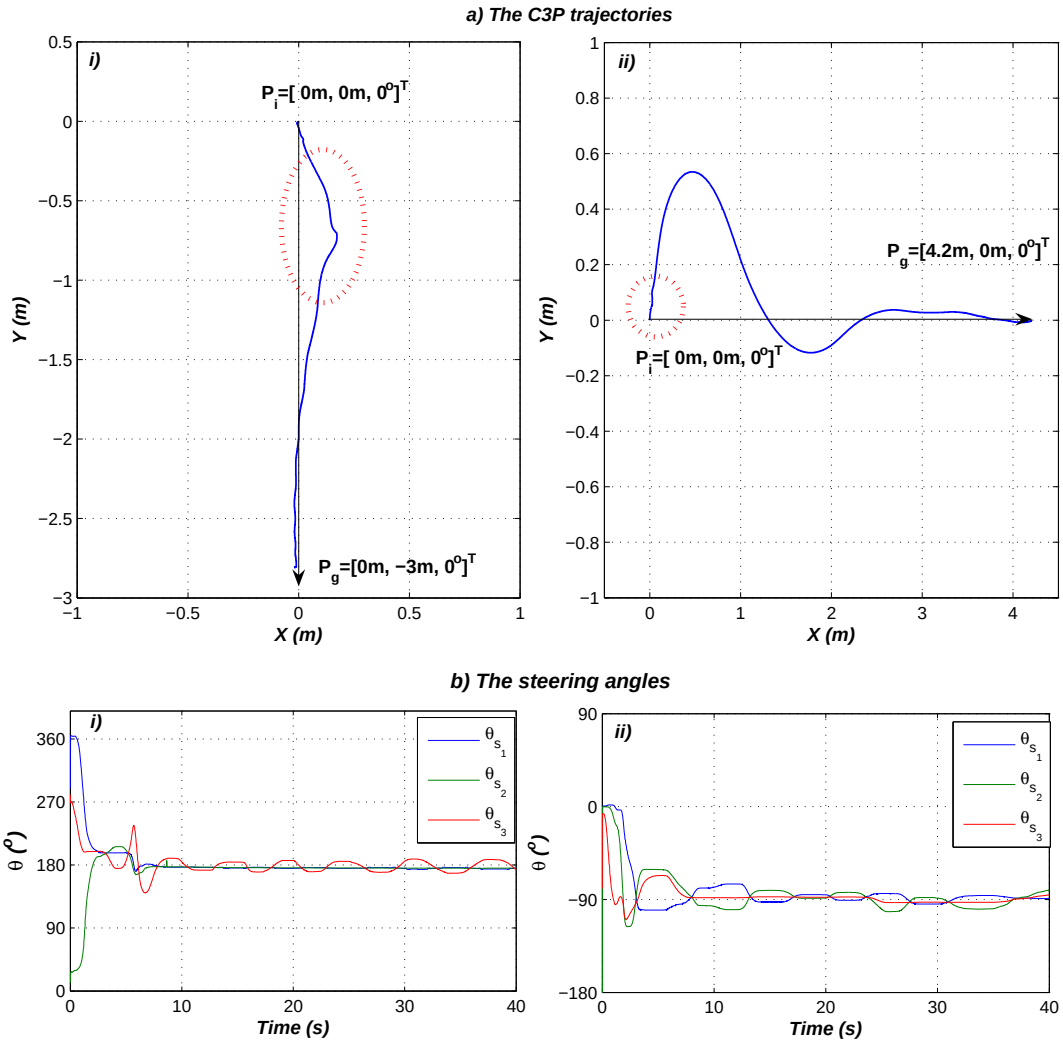

Figure 6.13. C3P trajectories and steering angles values for two experiments; a) driving to $\mathbf{p}_g = [0m \ -3m \ 0^o]^T$ and b) $\mathbf{p}_g = [4.2m \ 0m \ 0^o]^T$

6.3 Dynamic Based Control Results

The inverse dynamics based controller (Chapter 4) showed its feasibility in the simulation process. This section shows the robot behavior when the controller is implemented on the C3P prototype platform. Due to the slippage and the friction of the wheels, in addition to the errors between the C3P dynamic model and the practical prototype, the inverse kinematic and dynamic solution are used

in the solution. This solution is explained and discussed in Section 4.2.1, Figure (4.9). The 'Inverse Solution Dyn & Kin' block will be used in the velocity control structure shown in Figure (6.14)

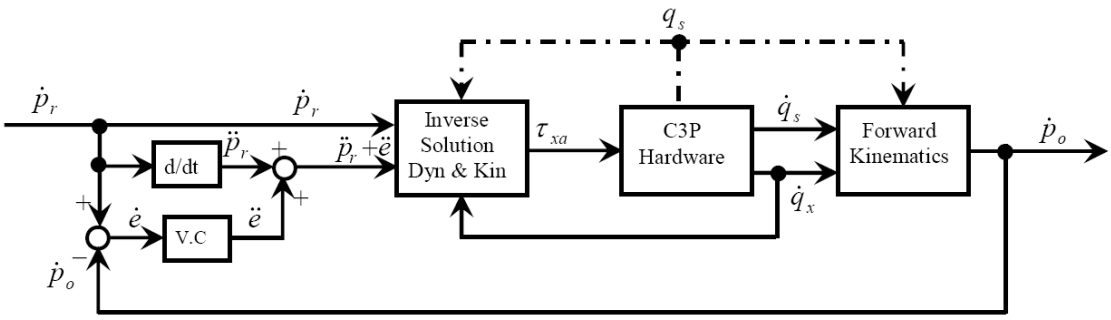

Figure 6.14. The C3P practical velocity control loop structure

The PID controller block in Figure (4.9)is the velocity axes control for each DC servomotor. The controller is implemented within the main program as a software controller. However, the wheel torque controller (T.C) is practically implemented in the motor control card BLD 5606. The BLD 5606 card has two built-in control loops: current and velocity control loops. The velocity control loop was used on the inverse kinematic motion control level. In this section the current control loop is used in Figure (6.15).

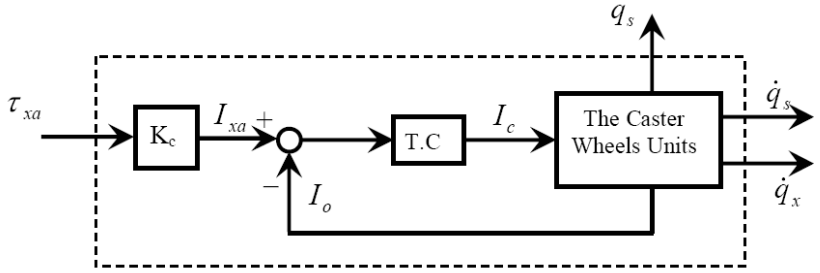

Figure 6.15. The C3P hardware block

The gain K_c is the torque current gain, which is defined in the data sheets. The caster wheel units sense two main variables; the wheel angular velocity vector and the steering angle vector. From the steering angle value the steering angular velocity vector is calculated; where $\dot{\mathbf{q}}_s = d\mathbf{q}_s/dt$.

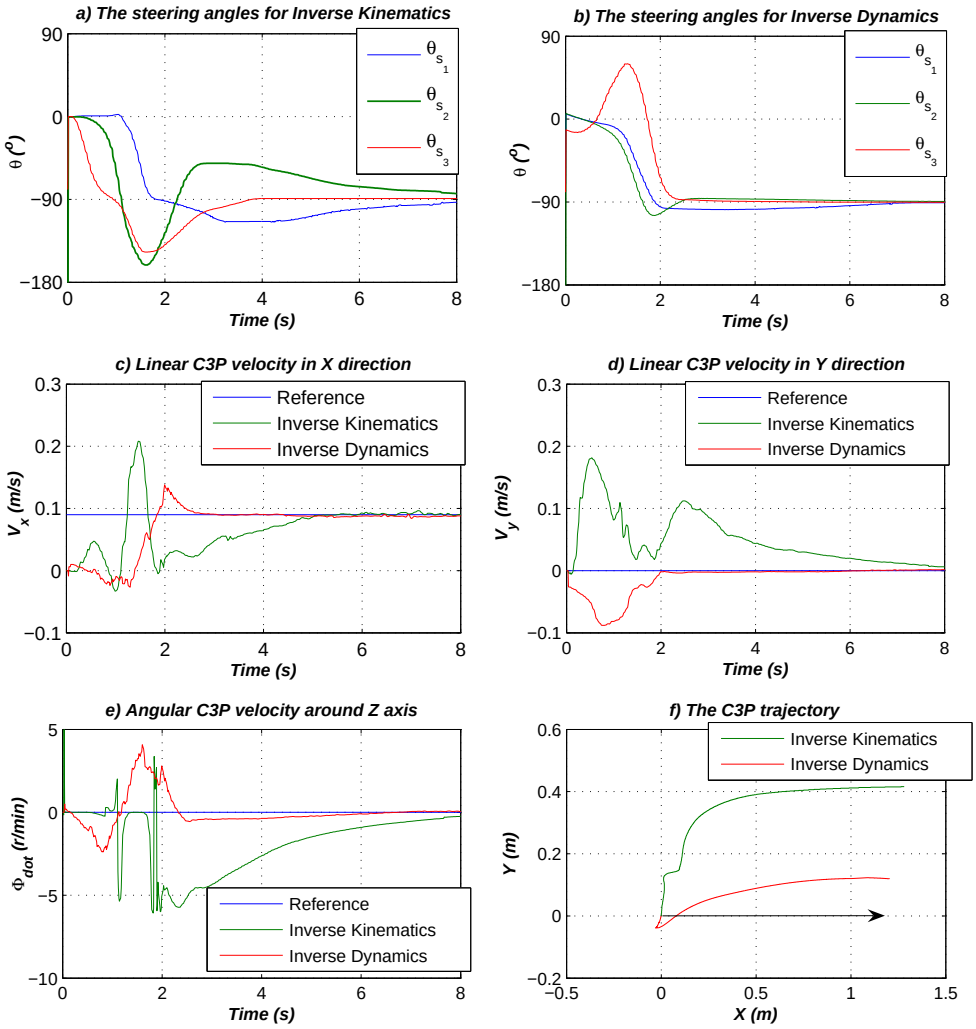

Figure 6.16. The C3P practical results from the inverse dynamic and kinematic solutions for driving in x direction

The first experiment is a comparison between the DBC and the KBC to illustrate the efficiency of both systems. The initial steering angles are nearly zero degrees and the desired robot velocity vector is $\dot{\mathbf{p}}_r = [0.095(r/s)\ 0(r/s)\ 0(r/min)]^T$. Figure (6.16) shows the output results, where the steering angles should all switch to -90^0 value.

The steering angles shown in Figure (6.16a & b) resulted from the inverse kine-

matic and inverse dynamic solutions. It is clear from the figure that the steering angles generated by the dynamic solution reach the desired value in almost two and half seconds, while the ones generated by the kinematic solutions take from seven to eight second to reach such a value. Such performance shows that the inverse dynamic solution is much more effective than the inverse kinematic solution on the robot velocity control level. This is demonstrated from the robot velocities in Figure (6.16c, d & e). The robot linear and angular velocities reach their reference values in almost two and half seconds (same as the steering angles), while the kinematic solution robot velocities take from six to eight seconds to reach the reference value. It is also apparent that the proposed solution generates velocities with only one overshoot and minimum magnitude in comparison with the kinematic solution. The robot trajectory resulted from the inverse dynamic solution drives in the desired direction with minimum displacement errors in comparison to the kinematic solution trajectory. The dynamic trajectory errors are 20% of the kinematic trajectory.

The steering angles have a smooth trajectory, which reaches the steady state with no disturbances Figure (6.16-b). Such behavior is not common for the C3P, where unexpected disturbances on the steering angles may result in oscillation in the robot velocities. For example, in the second experiment, which shows such disturbances during the motion in positive y direction (Fig. 6.17).

The initial steering angles values are around 180^o and it is required to drive with desired velocity of $\dot{\mathbf{p}}_r = [0(m/s)\ 0.5(m/s)\ 0(r/min)]^T$. The steering angles shown in Figure (6.17a) show disturbances in steering angle number 3 (θ_{s_3}) after reaching steady state value. Such disturbances are expected to occur after or during the switching from the inverse dynamics to the inverse kinematics solutions.

The disturbances influence the robot linear and angular velocities, as shown in Figure (6.17c, d & e). Such velocity errors and oscillations are integrated to obtain displacement errors in the robot trajectory (Fig. 6.17f). The robot drives in y direction and oscillates around the y axis with a displacment of 0.8 cm value. The oscillation errors are in the span of $\pm 2cm$ till it reaches steady state.

The third experiment shows smooth mobility in a positive x and y direction. The steering angles start from an initial value around 40^o and switch to the value of -45^o in order to drive in $\dot{\mathbf{p}}_r = [0.5(m/s)\ 0.5(m/s)\ 0(r/min)]^T$. The based

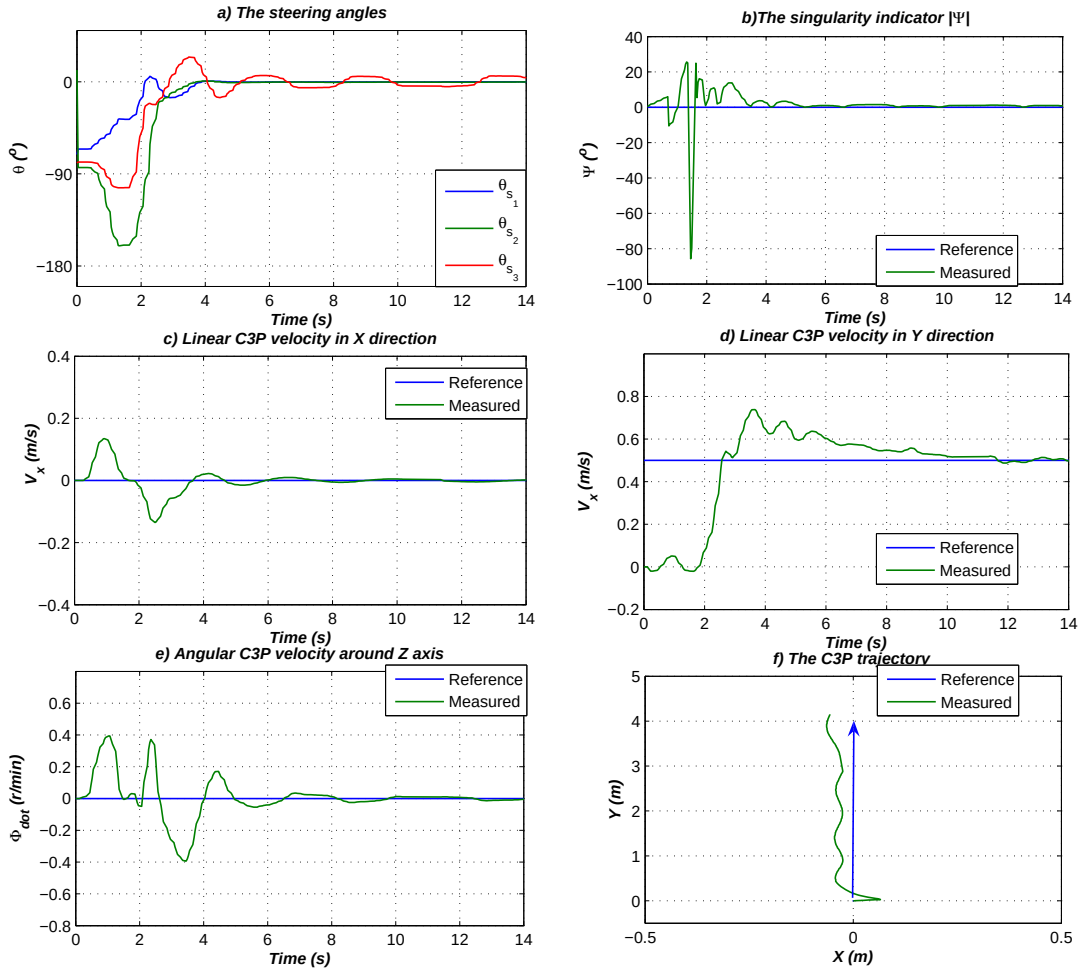

Figure 6.17. The C3P practical results from the dynamics based controller for driving in y direction

velocity controller succeeds in reaching the required robot velocities (Fig.6.18c, d & e). However, the positive oscillation in Figure (6.18-c) and the oscillations in the robot angular velocity in Figure (6.18-e) exist as a result of the steering wheel adjustment. Such velocity errors are integrated to influence the robot trajectory resulting shifting from the reference direction Figure (6.18-f). The red circle is the phase where the steering angles are adjusted. The black circle is the phase where the angles reachs the reference value and oscillates around it till they reach the steady state.

The next example shows the C3P performance while driving in infinity (∞)

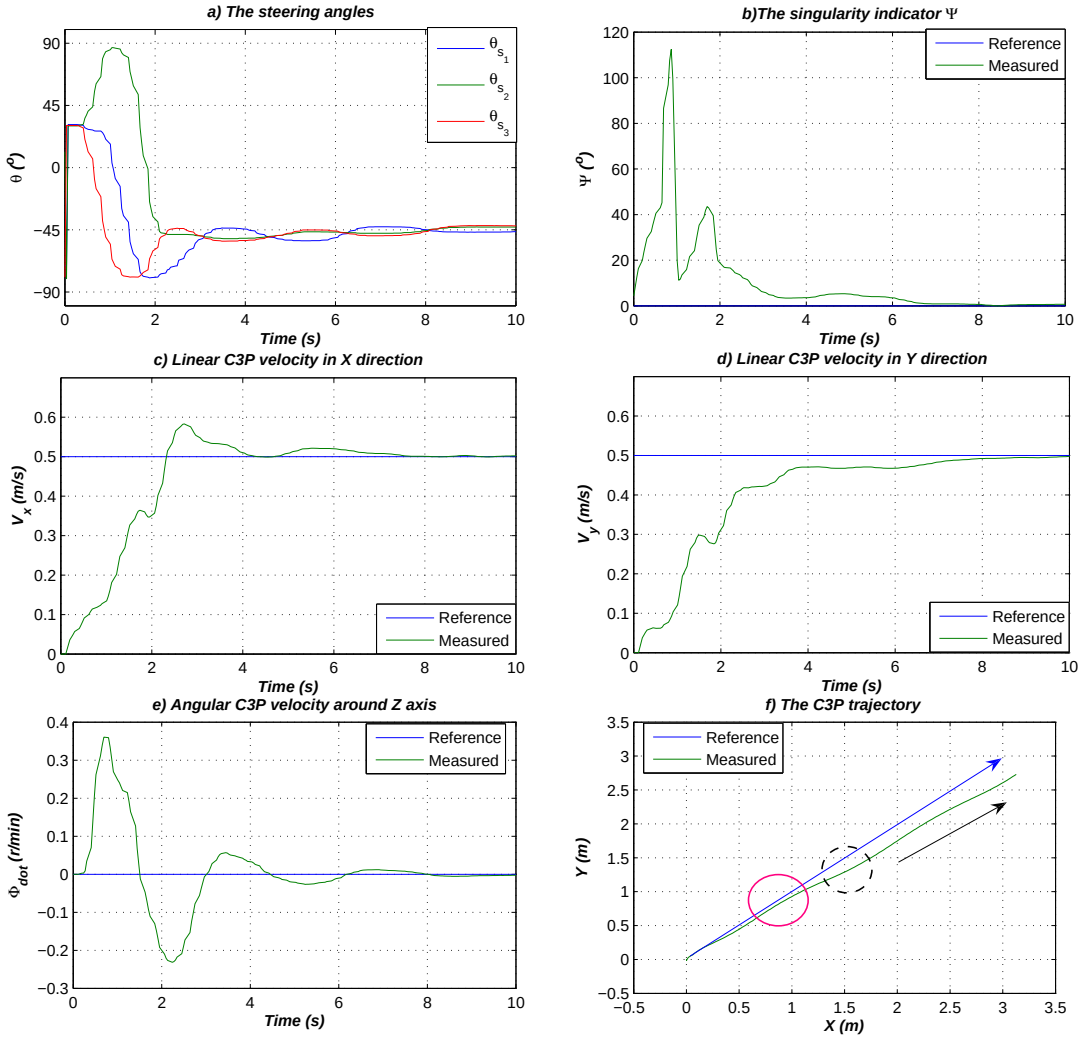

Figure 6.18. The C3P practical results from the dynamics based controller for driving in (x,y) direction

shape (Fig. 6.19). For smooth behavior of the used prototype platform, the robot will move in the y direction with respect to the robot frame of coordinates. Therefore the robot frame velocities are $\dot{\mathbf{p}}_{C3P} = [0\ 0.3(m/s)\ 0.2(r/min)]^T$ and floor frame velocities are $\dot{\mathbf{p}}_{Floor} = [0.3\cos(\phi)(m/s)\ 0.3\sin(\phi)(m/s)\ 0.2(r/min)]^T$. Such input robot velocities will drive the robot in a complete circle, when the C3P finishes the circle the $\dot{\phi}$ will be $-0.2(r/min)$ to drive in the other circle and complete the ∞ shape as shown in Figure (6.19). There are steady state errors

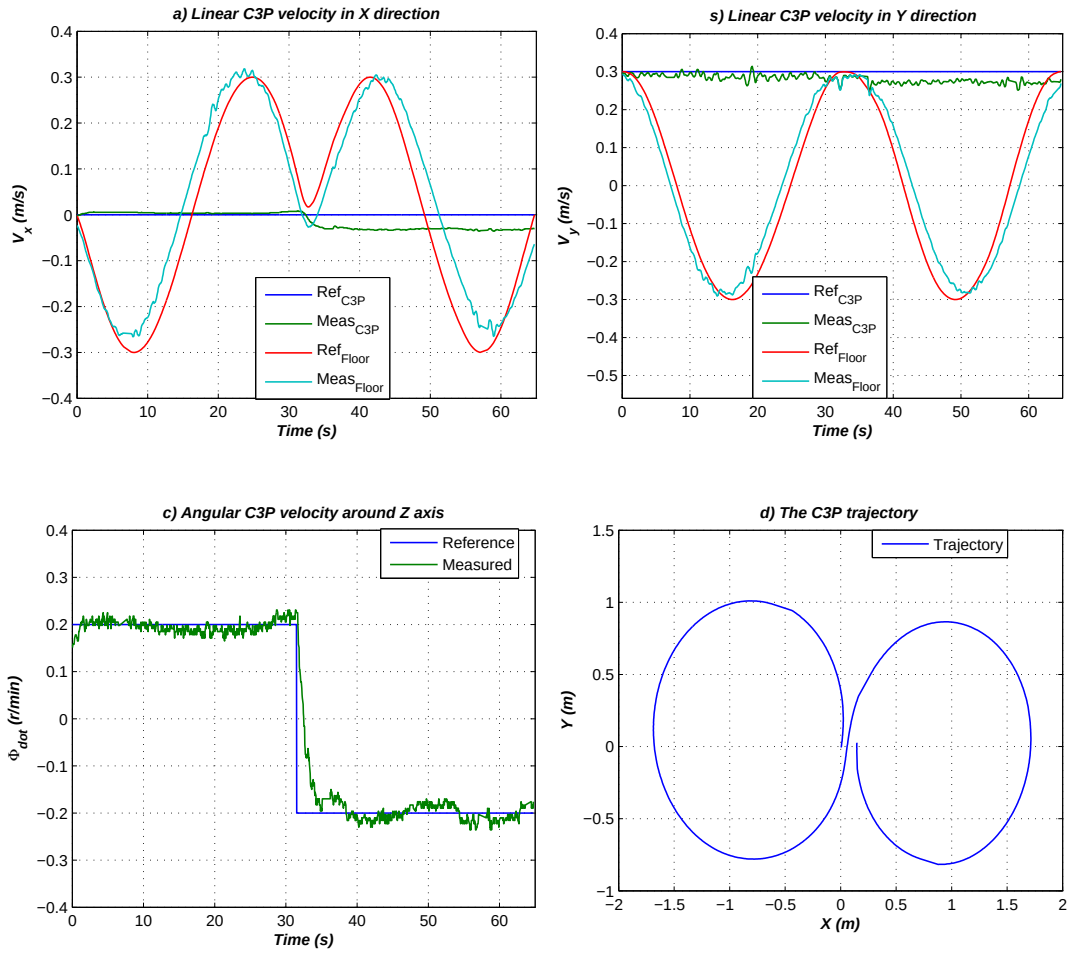

Figure 6.19. The C3P practical results for driving in ∞ shape

found in the floor frame robot velocities due to slippage and friction errors.

As previously explained, the C3P does not have direct actuation on the steering angles and the steering torques and the wheel friction and slippage are not included in the inverse Kinematic or the inverse dynamics solutions. That is the reason for the unexpected and unpredicted disturbances and oscillations found in the steering angles trajectories. The position controller discussed in Chapter (4) showed its effect and efficiency in the simulation process. The controller delivers robot reference velocities value to the velocity controller. The parameters μ_x and μ_y are set to the value '3' for better stability.

The first experiment for the position controller implementation is to drive from

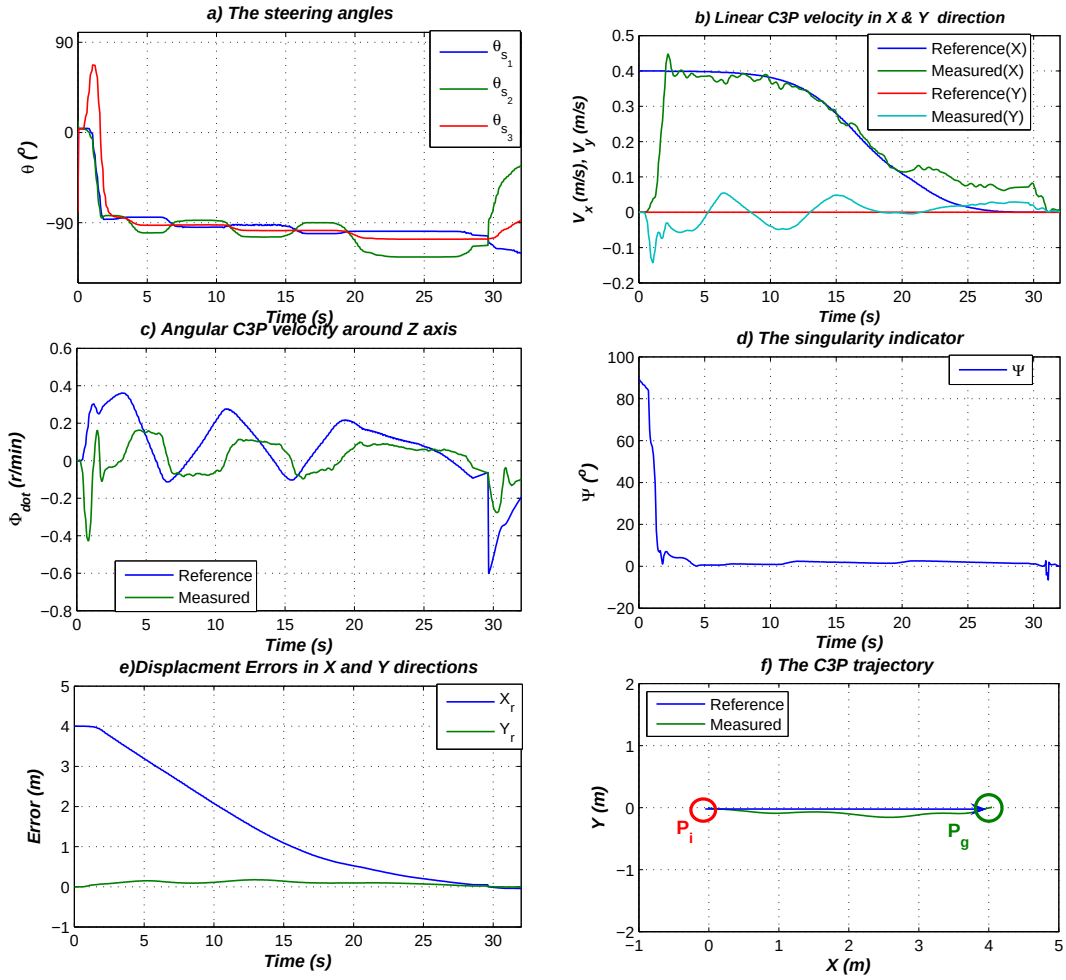

Figure 6.20. The position controller behavior for driving from $\mathbf{p}_i = [0\ 0\ 0]^T$ to $\mathbf{p}_g = [4m\ 0m\ 0^o]^T$

the initial point $\mathbf{p}_i = [0\ 0\ 0]^T$ to the goal point coordinate of $\mathbf{p}_g = [4m\ 0m\ 0^o]^T$. The initial steering angles are around the value of 0^o and the controller parameters are as follows: $K_x = 0.4$, $K_y = 0$, $K_\phi = 0.2$ and $K_{er} = -0.3$. The robot results are shown in Figure (6.20), where the steering angles took around two seconds to adjust their values to drive in the desired direction (x) (Fig. 6.20 a).

During the same period of time, the robot linear velocities reach their desired reference values. Figure (6.20d) shows the presence of a singular condition in the initial states of the experiment. After the wheel adjustment the system condition

became nonsingular. The goal point has a zero error area, since the practical prototype is hard to reach the exact goal point. This zero area is a circle with a radius of 5 mm, when the co-ordinates point of the robot reaches the robot velocities will be zero. From Figure (6.20b) it is clear that target goal zone is reached after 30 seconds. The robot trajectory is shown in Figure (6.20f), where the robot drives around the reference trajectory with error of ± 4 cm untill it reaches the goal zone.

The steering angles disturbances, which were explained previously in this section appear in the following example. The robot starts from almost -90^0 value for the steering angles and required to reach $\mathbf{p}_g = [0m\ 4m\ 0^o]^T$ goal co-ordinates with the following parameters $K_x = 0$, $K_y = 0.5$, $K_\phi = 0.25$ and $K_{er} = -0.3$. As shown in Fig (6.21b) the $\dot{x}$ or V_x velocity oscillates around its reference value which causes the robot angular velocity to generate the control signal plotted in (6.21c). Such control signal will generate disturbances in the steady state value of the steering angle number 3 (θ_{s_3}). The error signal coming from the variable δ_{er} is the main factor for such a signal, where it sensed that the robot is moving away from the goal zone. However, the signal succeeded in reaching the goal point in spite of the existance of friction and steering torque.

The odometry calculations used in the position control loop is a combination between the integration of the velocity errors (for the translational velocities $\dot{x}$ and $\dot{y}$) and an inertia sensor for the rotational angle value. In order to evaluate the performance of the position control, the C3P position update will be measured with more precise equipment. The device used for such measurement is Krypton motion tracker with 6 DOFs with a resolution om $5\mu m$ and a single-point accuracy of $50\mu m$. The previous experiment is applied once more with the same settings but for a goal point $\mathbf{p}_g = [0m\ 2m\ 0^o]^T$ (due to the limitation of the Krypton range). The robot starts with zero value for the steering angles. As shown in Figure (6.22) the data measured by the odemetry reaches the goal point with error of 5 cm, which is the zero zone. This trajectory is the one that the C3P reads and adjusts its position controller accordingly.

On the other hand these are not the real data, because of two main factors: a)the wheel slippages and b)incremental encoders errors accumulation. The Krypton trajectory shows the effect of such factors. This experiment was carried out six

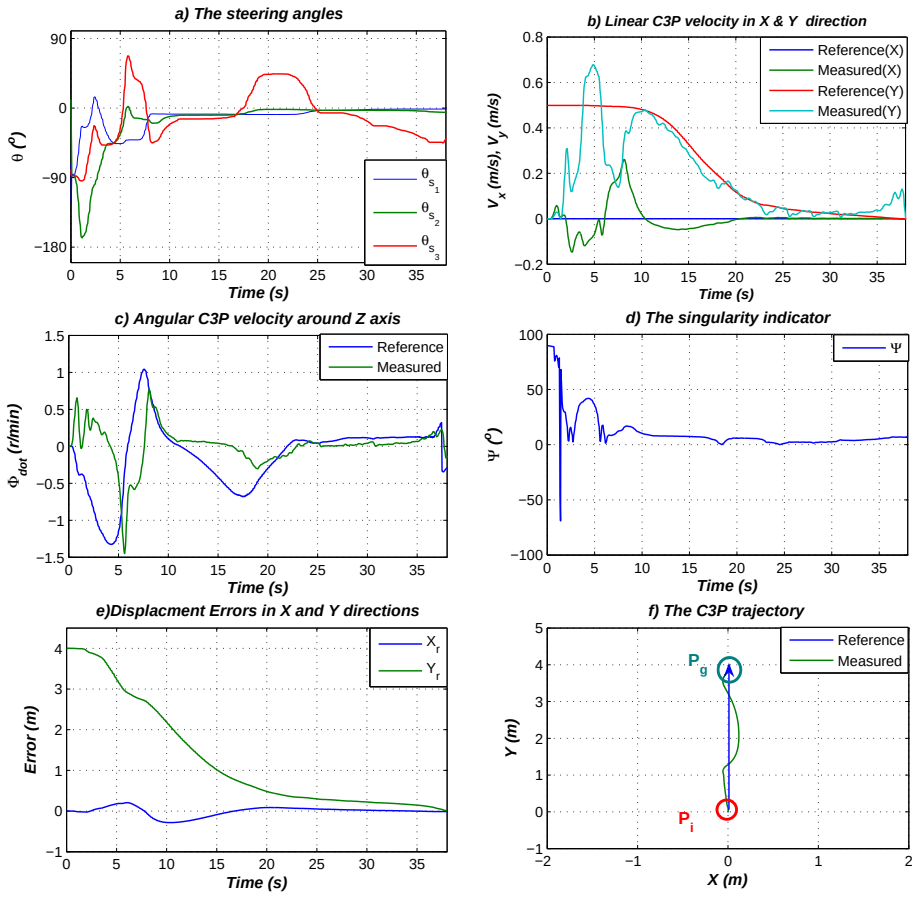

Figure 6.21. The position controller behavior for driving from from $\mathbf{p}_i = [0\ 0\ 0]^T$ to $\mathbf{p}_g = [0m\ 4m\ 0^o]^T$

times to use its data to check the performance of the controller. The figures only represent the trajectories of one experiment to help demonstrate the calculations.

First, the error between the point $\mathbf{p}_{Odem}$ (end point measured by the odomentry) and $\mathbf{p}_{Kryp}$ (end point measured by the Krepton)is calculated as follows: $X_{er} = X_{Odem} - X_{Kryp}$ and $Y_{er} = Y_{Odem} - Y_{Kryp}$, then the resultant distance error is $R_{er} = \sqrt{x_{er}^2 - y_{er}^2}$. The ten experiments resulted ten errors, which are used to calculate the percentage relative error (Rel_{er}), the mean (M_{ean}), and the standard of deviation (DIV)in the Table 6.2.

In the following experiment the Krepton and the Gyro sensors are also used for the trajectory measurements. The Gyro was used in the position control loop,

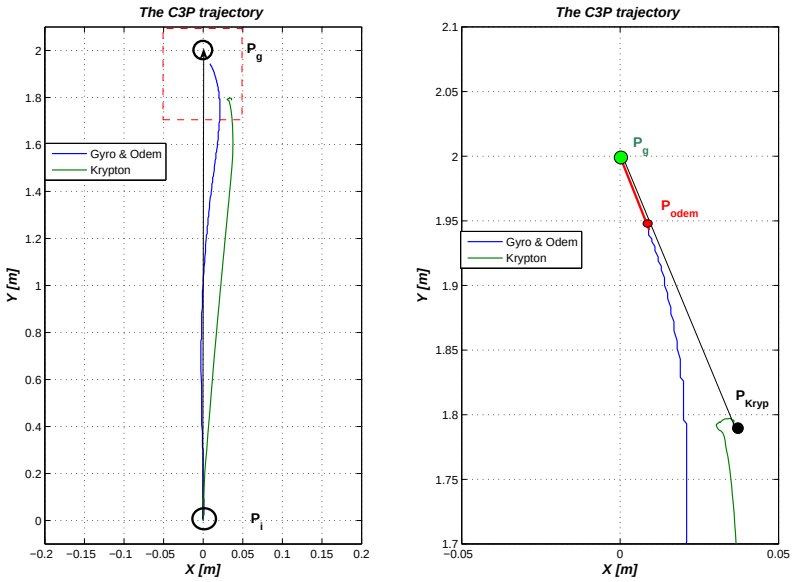

Figure 6.22. Comparing the Gyro & Odometry data to the Krypton data

while the Krepton was only used as a reference measurement. The C3P is set to drive in 1.8 meters in x direction, as shown in Figure 6.23.

In Figure (6.23-a)it is apparent that both trajectories are almost the same. By the zooming in Figure (6.23-b) the difference between the two measurements is quite clear. The Gyro measurements show that the C3P reached its Zero Zone (5 cm radius), while the Krepton shows that the C3P is out of the Zone with at least 12 cm due to error displacement on the y axis. To define the reason for such an error, the difference value between the Krepton and the Gyro measurements are calculated by the resultant displacement error(R_{er}). Figure (6.23-c) shows that the error is high at the beginning of the experiment. Such errors exist in each experiment due to the high slippage of the wheels at the transient state of the experiment. These errors decrease during the course of the experiment and accumulate as a steady state error.

As previously mentioned, the behavior of the C3P depends on its initial parameters, such as the initial steering angle values, the initial angular velocities and the initial steering angular velocities. Therefore, the robot should be tested with a discrete trajectory, like a triangle shape, as shown in Figure (6.24e) The exper-

Table 6.2. The position performance errors and evaluation

Exp #	R_{er}	$\%Rel_{er}$	Y_{er}	X_{er}
1	0.157 m	7.8%	0.155 m	0.024 m
2	0.051 m	2.5%	0.045 m	-0.024 m
3	0.092 m	4.6%	0.091 m	-0.0135 m
4	0.075 m	3.7%	0.06 m	0.045 m
5	0.122 m	6.1%	0.112 m	0.048 m
6	0.131 m	6.5%	0.13 m	-0.016 m
7	0.0658 m	3.1%	0.06 m	0.027 m
8	0.1215 m	6.0%	0.121 m	0.011 m
9	0.0636 m	2.7%	0.055 m	-0.032 m
10	0.0851 m	4.1%	0.083 m	0.019 m
M_{ean}	0.0964 m	4.8 %		
DIV	0.0347			

iment is divided into three stages. Stage one is to drive from $\mathbf{p}_1 = [0m\ 0m\ 0^o]^T$ with initial steering angles or zero degrees to $\mathbf{p}_2 = [0m\ 3.5m\ 0^o]^T$. Stage two is driving from $\mathbf{p}_2$ to $\mathbf{p}_3 = [3m\ 3.5m\ 0^o]^T$, and the final stage is to drive back to the starting point $\mathbf{p}_1$. This experiment illustrates the performance of the position controller in 3DOFs.

From the steering angles values shown in Figure (6.24a), the direction of the C3P can be easily observed: they are zero during the first stage (driving in y direction) then the values switch to -90^o (driving in x direction) then their values were around -220^o which represents driving in a (-x,-y) direction. But the values of the steering angles are sensed according to the C3P frame of coordinates. Therefore, the response of the linear velocities V_x or $\dot{x}$ and V_y or $\dot{y}$ (Fig. 6.24b & c) are sensed with respect to the floor coordinates. They show that the output velocities follow the reference signal with a delay of 17.5 seconds to 19 seconds because of the steering angle switching. Oscillations are noticed as well due to the floor friction, steering torques, and slippage. These three factors exist in the practical implementation, while they are absent in the inverse solutions and simulations. For this reason they often appear in a practical experiment. The C3P trajectory shown in Figure (6.24e) illustrates that the robot achieves its goal in the three stages. First, it reached point $\mathbf{p}_2$ with a final displacement error of 2 cm, and stage three with

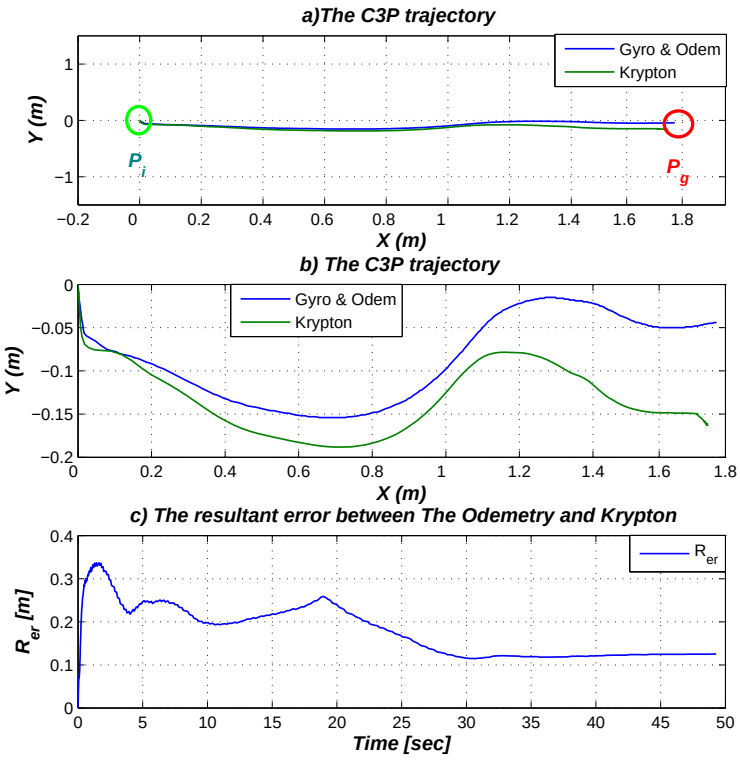

Figure 6.23. Comparing the C3P trajectory measured by the Krepton and the Gyro sensors

an error of 4 or 5 cm, and finally it reached the starting point again with an error of 10 cm.

6.4 Experiments on C3P Stability

It is quite important to study the stability of the position controller practically implemented on the C3P platform. In this section two experiments are presented, with both of them applying external disturbances on the steering wheel angles.

The first experiment is to drive the robot from the initial position $\mathbf{p}_i = [0\ 0\ 0]$ to the goal position $\mathbf{p}_g = [4\ 0\ 0]$. After the robot reaches its steady state condition (all steering angles are nearly equal) and is driving towards its goal point, at time =10 seconds an external disturbance is applied to wheel number three. The disturbance is defined by flipping the steering angle θ_{s_3} with 90 degrees, as shown in Figure 6.25-a.

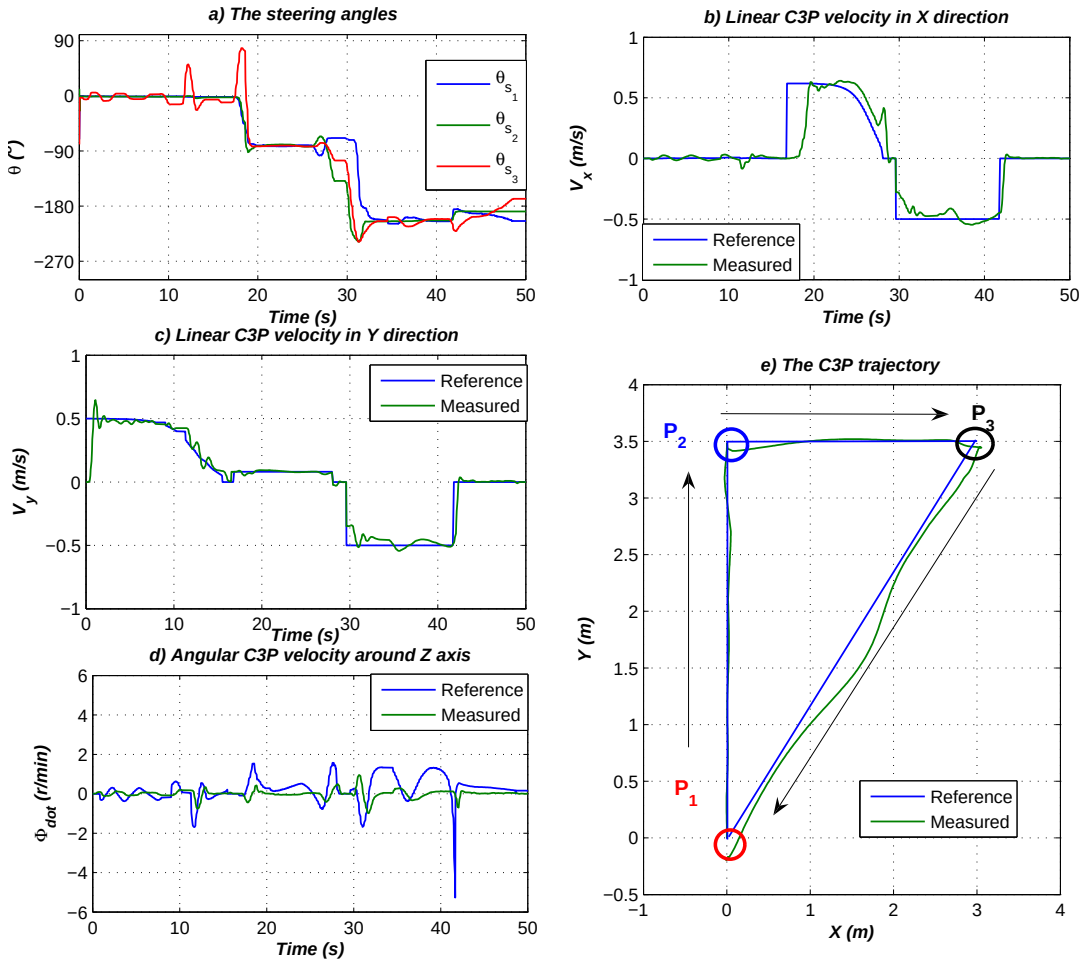

Figure 6.24. C3P behavior for driving in triangle shape

The controller managed to reduce the effect of these disturbances and brought the robot to its steady state again and reached the goal point after 20 seconds. The disturbance data was locked for all the states values and the Lyapunov function derivative $\dot{V}$ was calculated. The disturbance reached its peak value at a time of almost 10.7 sec. At this precise instant the locked data are the following:

The steering angular velocities $\dot{\mathbf{q}}_s$ have a zero value because at the peak value of the steering angles the first derivative of these angles is zero. Before and after the peak values, the resulted values of the Lyapunov function will include the value of the term $\rho_c^T(t)\ \dot{A}_v(\mathbf{q}_s, \dot{\mathbf{q}}_s)\ \rho_c(t)$.

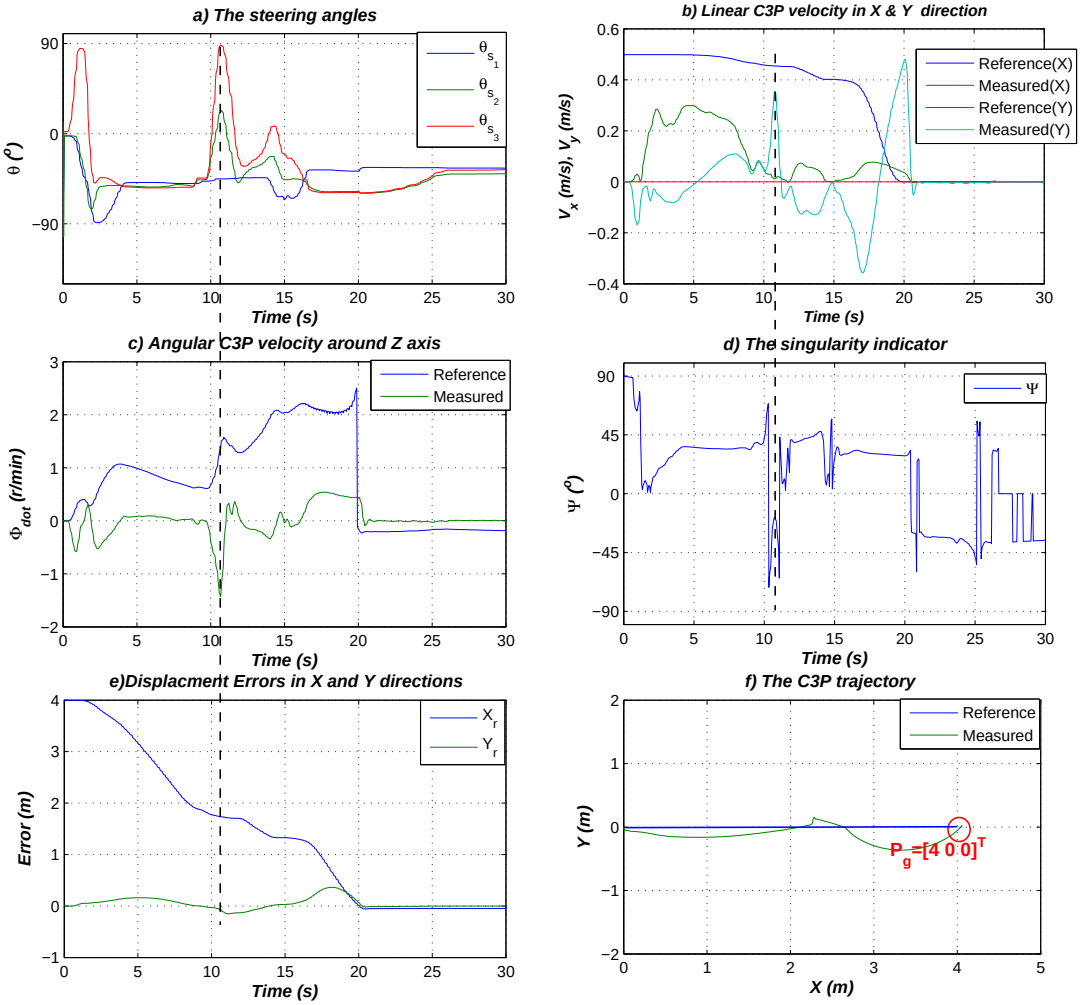

Figure 6.25. The C3P behavior for sudden disturbances in single steering angle

For the given values, the first derivative of the Lyapunov function is $\dot{V}$=-147, which means that the disturbances operated the system in a stable condition and the controller (with its parameter values) managed to bring the system to its steady state.

The second experiment operates with the same initial conditions but the disturbances are applied to the three steering angles at the same time, as shown in Figure6.26.

The disturbances are described in flipping the three steering angles by 90^o at

Table 6.3. The states values at high disturbances for Figure 6.25

K_x=0.5	K_y=0	K_ϕ=0.5
x(t)=2.2 m	y(t)=-0.1 m	$\phi(t) = 20^o$
$\dot{x}(t)$=0.35 m/s	$\dot{y}(t)$=0.02 m/s	$\dot{\phi}(t)$=1.2 r/min
$\theta_{s_1} = -45^o$	$\theta_{s_2} = 20^o$	$\theta_{s_3} = 90^o$

the same time, which drove the robot to instability and its trajectory kept moving in a circle, as shown in Figure 6.26-f. In this experiment the data are locked, as in the first experiment, at the moment of disturbances when the time is nearly 25 seconds.

Table 6.4. The states values at high disturbances for Figure 6.26

K_x=0.5	K_y=0	K_ϕ=0.2
x(t)=4.7 m	y(t)=0.2 m	$\phi(t) = -1.5^o$
$\dot{x}(t)$=0.06 m/s	$\dot{y}(t)$=0.0 m/s	$\dot{\phi}(t)$=-1.5 r/min
$\theta_{s_1} = 1^o$	$\theta_{s_2} = -2^o$	$\theta_{s_3} = 20^o$

For the given parameters, the first derivative of the Lyapunov function is positive ($\dot{V}$=66.2). The disturbances bring the robot to instability, while the controller could not bring the system to its steady state again. What would happen if one of the position control parameters value had been changed? For example, the parameter K_ϕ will be -0.2 instead of 0.2, then given$\dot{V}$=-19.5, which indicates that the system can be controlled. On the other hand, the value -0.2 shows the posibillity of instability for many steering angles combinations.

6.5 Summary

The practical prototype of the C3P was introduced in this chapter, along with its practical results. The output results showed the efficiency of the kinematics based controller with its velocity and position controllers. Differences between the practical and the simulation outputs are found. This is due to the absence of friction and slippage in the C3P forward dynamics model. The inverse dynamics

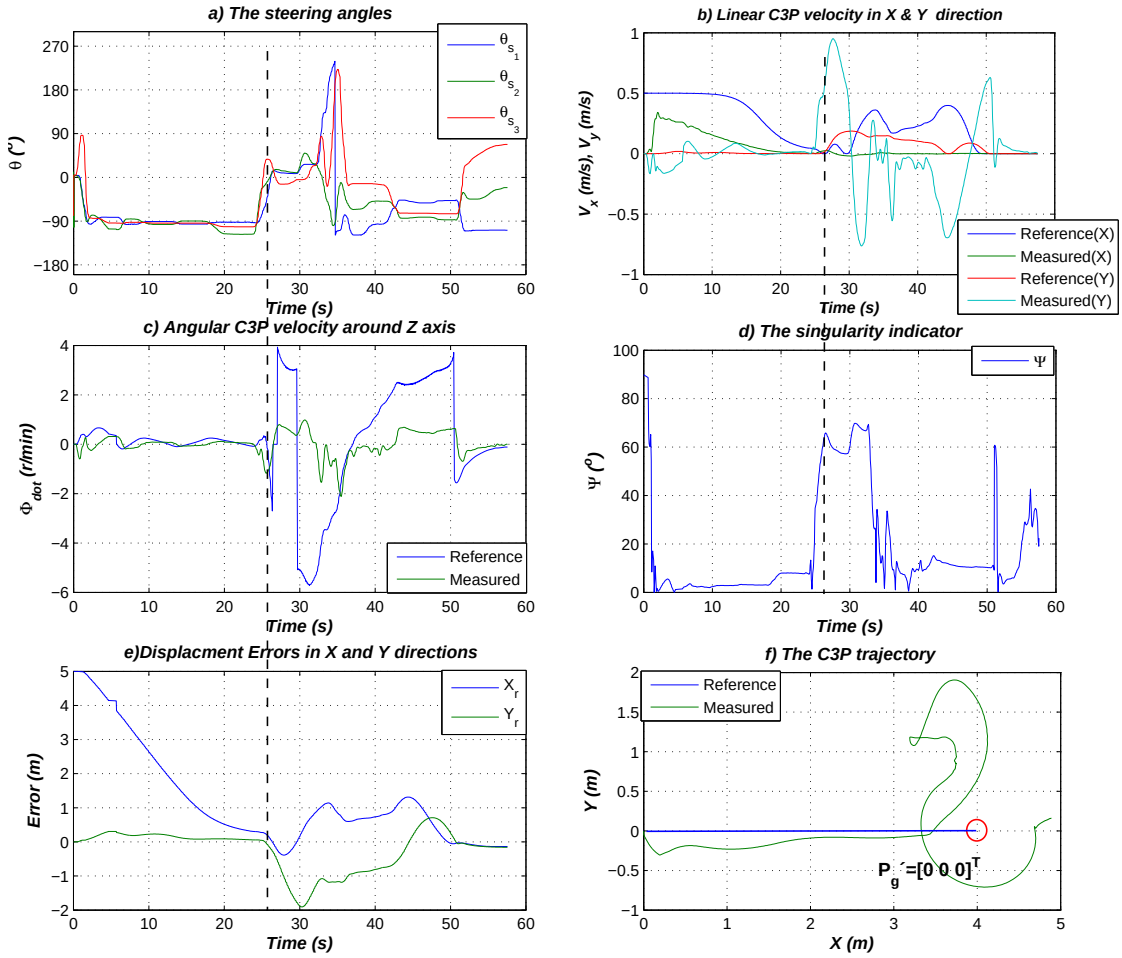

Figure 6.26. The C3P behavior for sudden disturbances in the three steering angle

based controller results shows that it has a better performance level than the kinematics one. The control structure used is simpler than the one used for the kinematics, which is illustrated in the velocity and position results.

Chapter 7

Comparing Different Holonomic WMRs

7.1 Introduction

The holonomic mobility of a WMR can be achieved by using wheels of 3DOFs motion. The wheels types that provide this mobility are: caster wheels, omnidirectional wheels, and ball wheels. Although the C3P carries the holonomic mobility features, it cannot be actuated in the direction perpendicular to the wheel angular velocity vector when the steering angles yield the same value. Thus, the actuation elements play an important rule in the WMR mobility. The C3P singularity problem was solved in Chapter 3 by the means of the coupling approach and special structural velocity controller.

In this chapter, a comparison between the C3P and three holonomic mobile robots is illustrated. The first robot is the **H**olonomic **C**aster **W**heeled **R**obot (HCWR), which is the same construction as the C3P, but with different actuation elements to insure its holonomic motion. The second WMR is an **O**mni **D**irectional **W**heeled **R**obot (ODWR) [98] [99], with three omnidirectional wheels to give the ability of free rotation in one direction, and powered rotation in an orthogonal direction [100] [101]. The third was proposed by the Swiss Federal Institute of Technology (ETH) under the name of RAMSIS II [102] and consists of two parts: a driving platform with 2DOF mobility connected to the second part with an actuated turret to provide the third DOF.

To be able to evaluate different robot platforms with respect to certain criteria, a performance criterion has to be set up for each robot platform [103]. Much work has been done in this field by the Robotics Research Group at the University of Texas (Austin); see for example [104], [103], [105], and [106]. In [103], several performance criteria for different aspects are developed, such as joint level constraints, transmissibility, kinetic energy, deformation, or energy distribution. However, a detailed literature review showing a criterion for describing the structure or hardware complexity of a robot configuration has not been found.

7.2 Description of Holonomic Mobile Robots

The kinematic model is commonly used as the primitive stage of WMR velocity control due to the following: a) it contains most of the geometric constraints of the platform including the wheels configuration [108], b) it is applicable to any type of planner mobile robots [109], c) the mobility and actuation characteristics can be easily concluded from the kinematic model of a WMR [4]. The forward and inverse kinematics will be obtained from the equations discussed previously in Chapter 2.

The configuration of the Holonomic Caster Wheeled Robot (HCWR) platform has the same geometric parameters of the C3P with three caster wheels. The coordinate systems of the robot are assigned in Figure7.1, where the wheel radius r is 0.03 m and the offset link d is 0.04 m. The distance between the robot geometric center and each hip h is 0.343m. The main difference between the C3P and the HCWR are the actuation elements; the HCWR has wheel angular velocities and the steering angular velocities as the actuated elements, whereas the C3P only has the wheel angular velocities actuated.

The omnidirectional wheels have become popular in the WMR field due to their holonomic mobility features. The Omnidirectional Wheel is based on the following concept: while the wheel drives in the direction normal to the motor axis, the wheel can slide frictionless in the motor axis direction [110] [111].

The Omnidirectioanl Wheeled Robot (ODWR) used in this thesis is shown in Figure 7.2, where the geometric parameters of the robot are set to be as close as possible to the C3P geometric dimensions; r= 0.01 m (the radius of the roller), R=0.08 m (the radius of the wheel), and h= 0.343 m (distance between the plat-

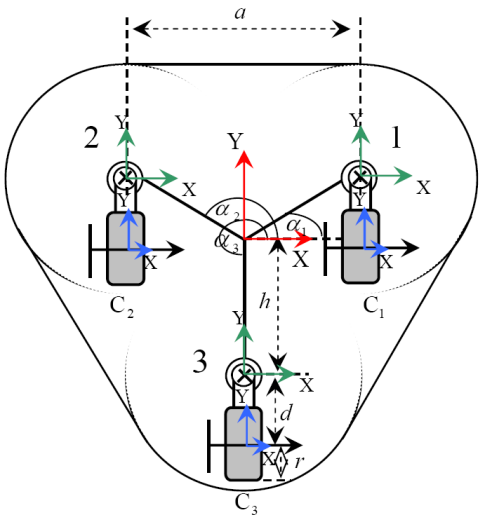

Figure 7.1. HCWR Configuration Structure

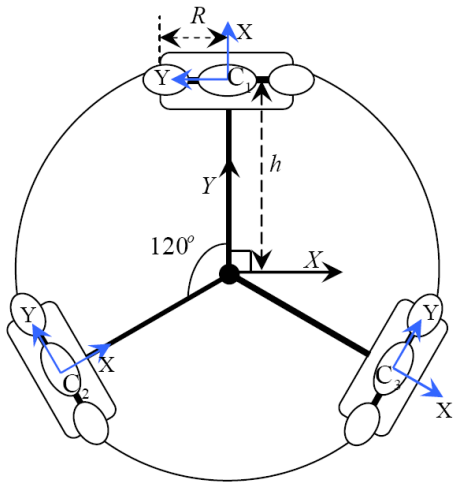

Figure 7.2. Omni Configuration Structure

form co-ordinates of reference and each contact point). The angle shifted between each wheel is symmetric to 120^o. Hence, the value of $\theta_1 = 90^o$, $\theta_1 = 210^o$, and $\theta_3 = 300^o$ while $\eta_i = 90^o$ for $i \in \{1, 2, 3\}$.

Ramsis II is the name of the mobile robot developed at the Automatic Control Laboratory of the Swiss Federal Institute of Technology (ETH) in Zurich [102]. The

basic configuration of the mobile platform is shown in Figure 7.3. The platform consists of two circular shaped plates mounted above each other with roller bearings allowing for relative rotation on an axis perpendicular to their surfaces. The lower part is a standard 2DOF mobile robot with two conventional 2DOF wheels apart from each other with the distance $2a$ where $a = 0.25\ m$. The 2DOF platform drives in the y direction and rotates around its mass of center in the black color. The upper part is the turret part which rotates around the z-axes of the main robot coordinates located away from the wheels steering angles with the distance b where $b = 0.15$ and r is the wheel radius.

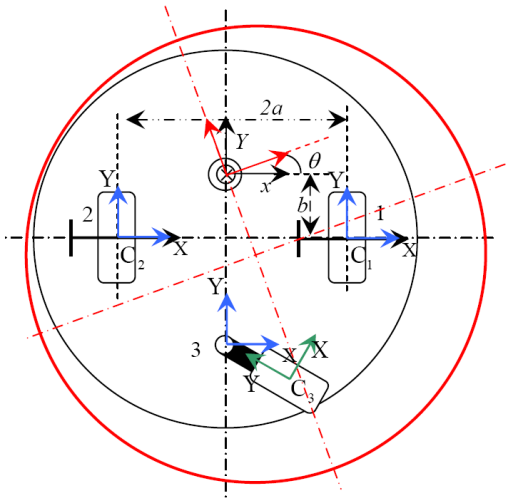

Figure 7.3. RAMSIS-II Configuration Structure

The turret part is presented in the the color(Fig. 7.3) with offset b, which provides the third DOF.

7.3 Comparing The C3P Vs Holonomic WMRs

The comparison illustrated in this section will be established with the simulation package carried out under the Matlab environment. The dynamic models needed for such a simulation are obtained in section (E.2) appendix (E) for holonomic WMRs. The inverse and forward kinematic solutions are shown in detail in Section

(E.1) appendix (E). The obtained models will be used in the simulation structure shown in Figure7.4.

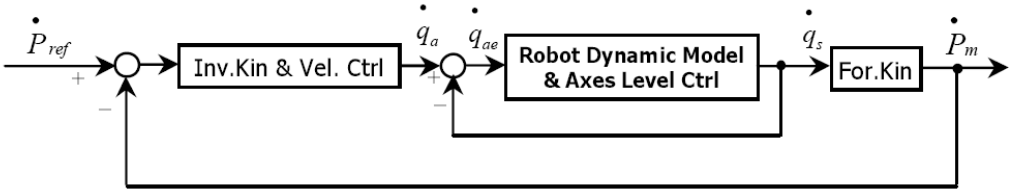

Figure 7.4. Velocity control loop structure

The structure describes the WMR velocity and the axes level control loops. The axes level loop are the actuators controllers which are regular PID controllers for DC motors, since the DC motors are the only actuators used in the three WMRs and the C3P. The WMR velocity controller can be different in structure and configuration in the accordance with the mobility constraints. The WMRs used in this section have holonomic actuation characteristics, hence a regular PID controller will be used with a different parameter setting with regard toeach WMR.

The two main items used in such a comparison are: the robot physical construction and simulation results. The physical construction will show the complexity in construction and actuators elements compared to the C3P. The simulation results focus on three main results: a) the robot error trajectories, b) the robot velocity response, and c) the energy consumed to drive the robot in a certain direction between the time interval t_1 and t_2, which can be calculated first from the power equation

$$\mathbf{p}_i(t) = \tau_i(t)\dot{\mathbf{q}}_i(t), \tag{7.1}$$

where $\tau_i(t)$ and $\dot{\mathbf{q}}_i(t)$ are the torque and angular velocity of actuator i. Second, the robot total energy is the sum of the actuator's calculated powers

$$P_T(t) = \sum_{i=1}^{N} P_i(t), \tag{7.2}$$

where n is the number of actuators. Finally, the total energy consumed by the

robot during finite time interval is described as follows

$$E_T = \int_{t_1}^{t_2} P_T(t)\ dt, \tag{7.3}$$

In this chapter the four robots have the same mass, similar dimensions, and shape. Therefore, the energy calculations are considered to be per unit mass. The energy consumption depends mainly on the dynamics of the model, which are the robot velocities and accelerations. There are two examples with this comparison: a) driving in 3DOF with a step velocities input, and b) driving in infinity shape (∞), which provides continuous velocities and accelerations.

7.3.1 Driving in 3 Degrees of Freedom

Driving in 3DOFs is sufficient to illustrate the robot performance while driving with constant velocities and zero accelerations. This section presents the behavior of the four holonomic robots when driving in 3 DOFs with the input velocity vector $\dot{\mathbf{p}}_{ref} = [0.12(m/s)\ 0.12(m/s)\ 1.9(r/min)]^T$, starting from an initial zero velocity. Figure(7.5) represents the simulation outputs resulting from the mentioned example, where the omnidirection wheeled robot (ODWR) consumes the highest amount of energy (Fig.(7.5a)) because the omnidirectional wheel requires high energy to role in incline direction to its angular velocity axis. In addition to the motor's high current needed to overcome the wheels and the robot torques to drive in 3DOF, which is less in case of driving in 2DOFs or 1DOF. The HCWR in second place due to the energy consumed by the steering angular velocities actuators, which do not exist in the C3P. Ramsis II comes in third place, due to the energy consumption in the turret steering actuator. To conclude, the C3P has the best energy performance of the four platforms.

The linear velocities $\dot{x}$ and $\dot{y}$ (V_x and V_y) (Fig.7.5b& c) show that the C3P velocities oscillate before reaching the desired value. The resulted oscillations have a high magnitude in comparison to the other robots. The omnidirectional wheeled robot reaches the reference value smoothly in one second. The angular velocity shown in Figure (7.5d) also demonstrates the oscillations resulted in the C3P response. The robot velocities behavior affects its position performance; therefore

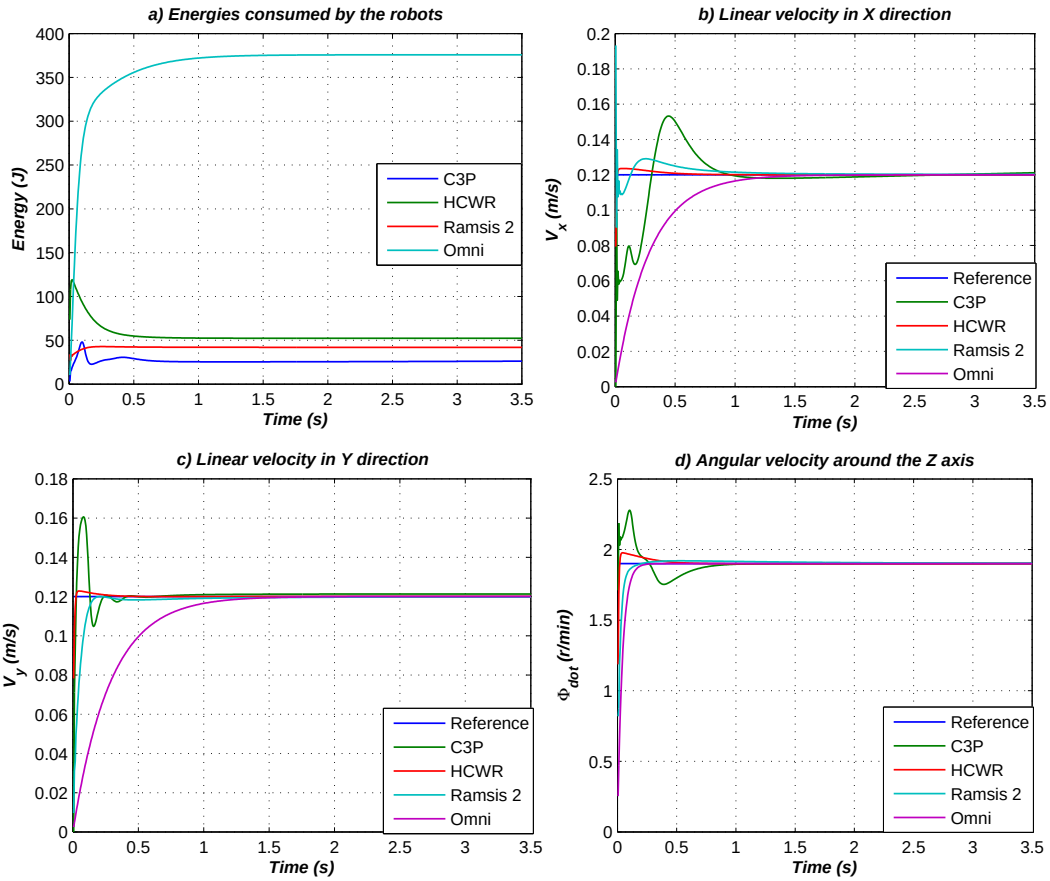

Figure 7.5. Energies and Velocities for Driving in 3DOF

the integration of the velocity error is presented in Figure (7.6) as the robot trajectories with respect to the floor frame of coordinates.

The C3P transient phase appears within the dashed circle, where the C3P controller succeeds in driving the robot in the desired direction but not on the reference trajectory like the HCWR and the ODWR robots. A similar trajectory error appears in the Ramsis II trajectory behavior.

7.3.2 Driving in the Infinity Shape(∞)

This section presents the four robot's performances while driving with variable and continuous velocities and accelerations. For this example, the shape of infinity (∞) is used as a trajectory with the following equations:

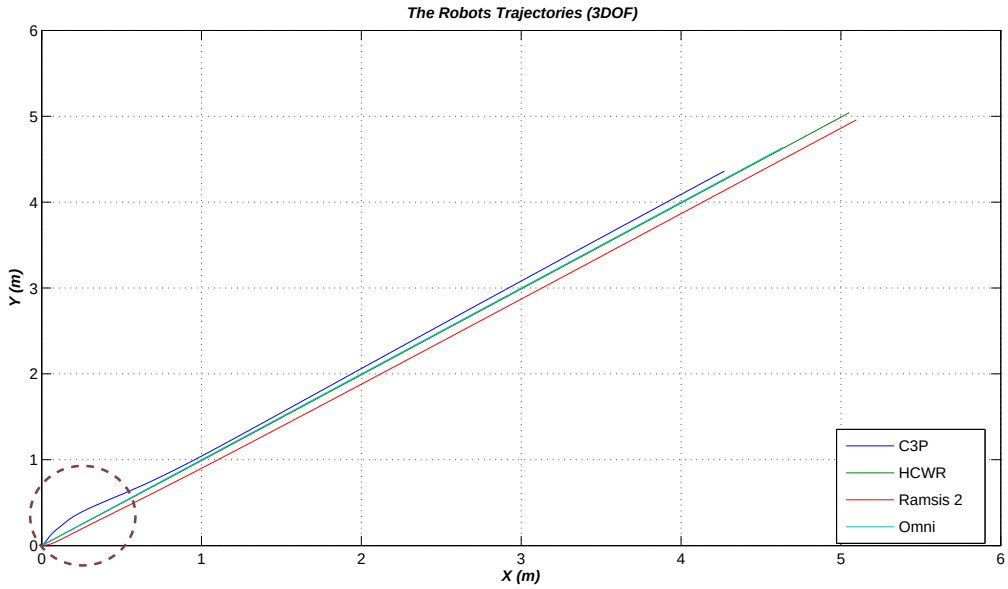

Figure 7.6. Robots' Trajectories for Driving in 3DOF

$$\dot{x} = d/dt \left(\frac{R_\infty \cos(\theta_\infty)}{1 + \sin^2(\theta_\infty)} \right), \tag{7.4}$$

$$\dot{y} = d/dt \left(\frac{R_\infty \cos(\theta_\infty) \sin(\theta_\infty)}{1 + \sin^2(\theta_\infty)} \right). \tag{7.5}$$

Figure(7.7) represents the simulation outputs resulting from the mentioned example, where the omnidirection wheeled robot (ODWR) consumes the highest amount of energ (Fig.(7.5a)) but varying with the shape of the curve depending on which wheel is actuated. The HCWR comes in second to the energy consumed by the steering angular velocities actuators, which do not exist in the C3P. Ramsis II comes in third place, where its energy is mostly consumed by the wheels motors beside the turret motor to keep zero angle orientation all through the trajectory. The C3P comes in fourth place, since the trajectory is continuous, in some intervales the system reaches singularities and the C3P requires high energy to get out of singularities and maintain the reference velocities. Examples for such periods are at Time=1.75 seconds. As a conclusion, C3P has the best energy performance of the four platforms.

The linear velocities $\dot{x}$ and $\dot{y}$ (V_x and V_y) (Fig.7.7b& c) show that the effect

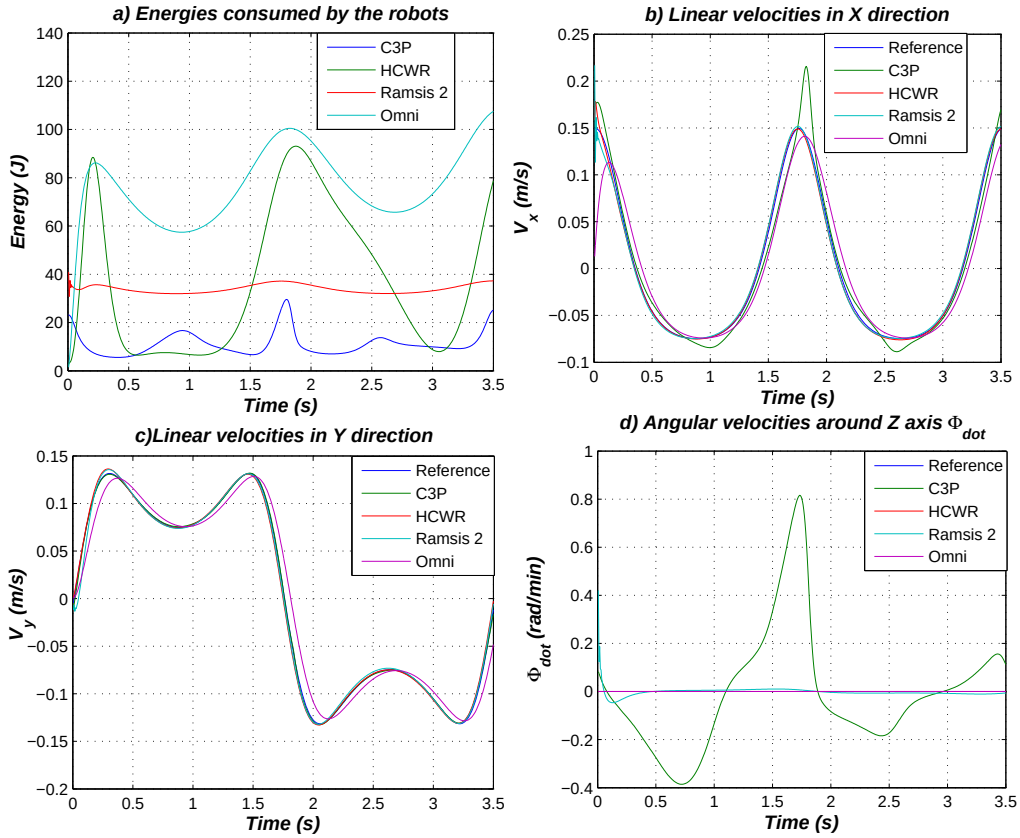

Figure 7.7. Energies and Velocities for Driving in (∞) Shape

of singularities conditions on the C3P responses. The same figures also show the shifting of the omnidirectional robot velocities from the reference value, yet it maintains similar trajectory. The angular velocity shown in Figure (7.7d) also demonstrates the oscillations resulted in the C3P response, where the C3P is the only robot with noticeable oscillations in the angular velocity. This is due to the absence of dynamic parameters (mass, inertia, friction, etc.) in the inverse kinematics solution. The robot velocities behavior affects its position performance, therefore the integration of the velocities error is presented in Figure (7.8) as the robots trajectories with respect to the floor frame of coordinates.

The most noticeable trajectories are the C3P and Ramsis II. Their trajectories are shifted away from the reference one with about 20 cm. This is due to the velocities errors found mainly in the x direction. The Ramsis II robot as transient

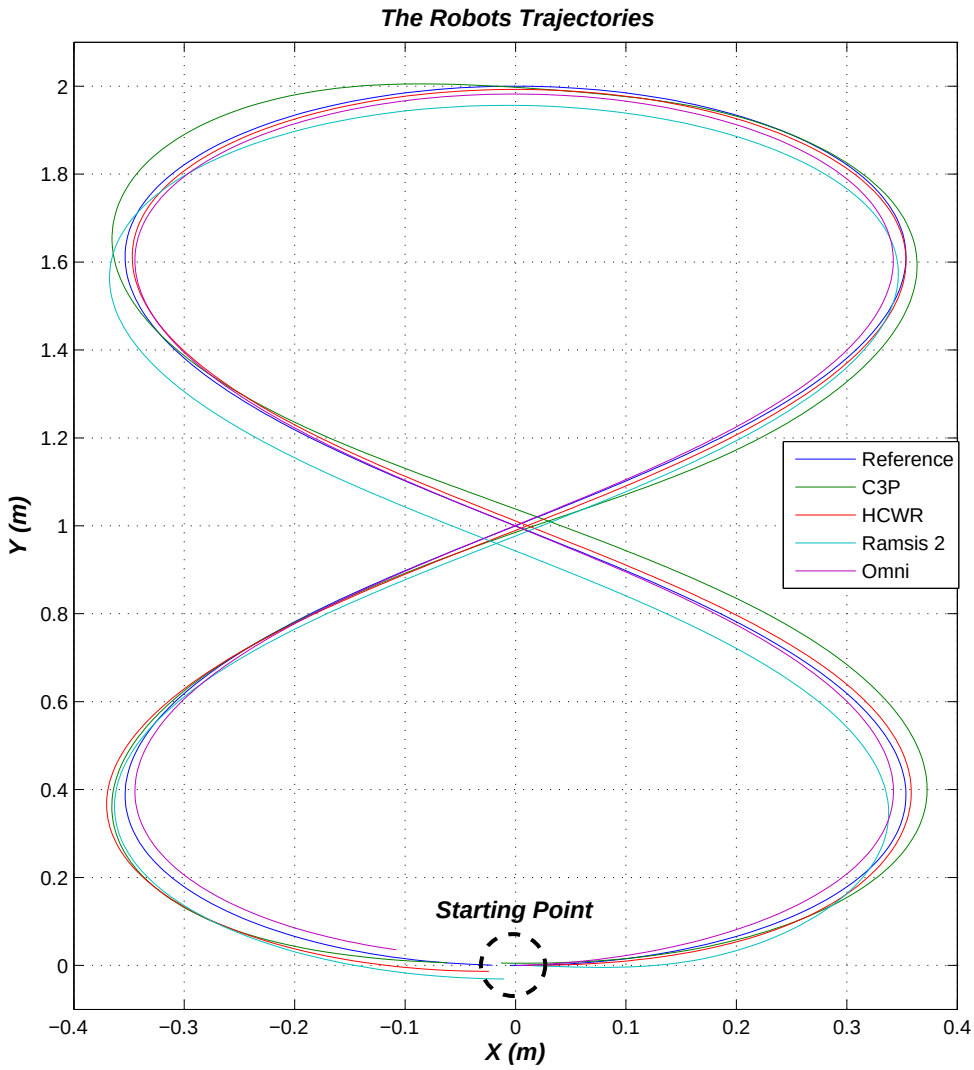

Figure 7.8. Robots Trajectories for Driving in (∞) Shape

oscillation in the angular velocity (Fig. 7.7d) which shifted it from the reference trajectory all through the rest of the example. The best robot for maintaining almost zero errors all over the trajectory is the omnidirectional wheeled robot. The trajectory errors are to be expected due to the absence of a tracking controller (combined velocity and position control).

7.4 Performance Function Comparison

This section illustrates the second part of the comparison, which is calculating an overall cost for each robot. The overall cost functional J_r for every robot is based on three main aspects: mobility, energy consumption, and hardware complexity. These three aspects are combined in the cost functional J_r using the weighted sum method [107][104]. The weighted sum method is a quantitative method for handling cost functionals with more than one functional [116] [117].

7.4.1 Mobility Aspect

For the calculation of the mobility cost functional C_m, the integration of the velocity errors in a finite interval of time T is used in this comparison. The mobility cost functional is split in two, cost functional (C_{m_l})for linear velocities and cost functional (C_{m_r}) for the rotational velocity described by the following equation:

$$C_{m_l} = \int_0^T \left| \sqrt{\dot{x}_{er}^2(t) + \dot{y}_{er}^2(t)} \right| \, dt, \tag{7.6}$$

$$C_{m_r} = \int_0^T \left| \dot{\phi}_{er}(t) \right| \, dt. \tag{7.7}$$

The total value of the mobility cost is the sum of their linear velocities $C_m = C_{m_l} + hC_{m_r}$, where h is the platform radius.

7.4.2 Energy Consumption Aspect

To set up the cost functional for the energy C_e, the total energy consumed instantaneously from driving the robot in a certain direction between the time interval t_1 and t_2 specified in equation (7.2), is considered [104]. The integration of the absolute value of equation (7.2) yields to the energy cost functional C_e, formulated as follows:

$$C_e = \int_0^T |E_T(t)| \, dt. \tag{7.8}$$

7.4.3 Hardware Complexity Aspect

The hardware complexity has a special cost function evaluation, since there is no specific known functional calculating method for its cost value. The mobile robot complexity can be determined according to the types and number of sensors, actuators, and wheels included in the robot construction. Generally, the active elements, parts, and joints are considered. These elements are classified as the following:

Electrical Elements : DC motors, Sensors.

Active Parts : Body, Links and Joints.

Wheel Unit : Caster, Conventional and Omnidirectional wheels

The robot platforms used in this chapter only have rotational actuation, which requires a DC motor for actuation and incremental encoders for velocity sensing or absolute encoders for absolute angle measuring. The active parts are classified into four parts: a) robot body, which has all the wheels and joints connected, b) joint, which is the connection between a rotatory element and another body, c) link, which connect two different joints. Three wheel units are used: a) the caster wheel, which has two joints, one link and one wheel, b) the conventional wheel has only one joint and one wheel, c) the omnidirectional wheel is the most complex wheel among them, which has one joint, one wheel, and six rollers mounted around the main wheel.

Each element of the above mentioned is considered as one complex unit presented with the letter 'x'. Also according to its number of DOFs; for example the joint has 1DOF (rotation), the motor has 1DOF but the body has 3DOF. Table 7.1 shows the complexity evaluation of each robot.

According to the table, each robot complexity value is described as the following: C3P has three incremental encoders on the axes of each DC motor to sense the wheel angular velocity, and three absolute gray encoders to sense the steering angles. There are three DC motors used in the C3P for the wheels angular velocities actuation, three caster wheels, and three bearing units.

The ODWR has three incremental encoders, three DC motors, no bearing units, and three omnidirectional wheels with the following distribution: each wheel is

Table 7.1. Robots Hardware Complexity Value

Elements / Platform	C3P	HCWR	ODWR	RAMSIS-II
Motor	3x	6x	3x	4x
Sensor	6x	6x	3x	5x
Link	3x	3x	0x	1x
Joint	6x	6x	3x	7x
Wheel	3x	3x	3x	3x
Roller	0x	0x	3*6x	0x
Body	1x	1x	1x	2x
Sum	**21x**	25x	**31x**	22x

assumed to have six rollers on its surface as the ones used in [118]. The HCWR is more complex than the C3P platform becuase of its three steering actuation units on each caster wheel, which means extra 3 DC motors.

The Ramsis II is considered a special case for its unique configuration. It consists of two bodies; upper and lower. The lower body is the fixed base and the upper body is actuated by a turret, which requires one bearing unit and two motors acting from both sides to apply coupling actuation [102] to maintain efficient steering. In addition to this it also requires two incremental encoders, two DC motors, two conventional wheels, and one caster wheel.

7.4.4 Cost Functional Calculation

Considering the different cost functionals in one functional using the weighted sum method (WS), the cost functional values have to be of same unity or without units. In this paper, each aspect cost value will be normalized depending on the maximum value among the four platforms [104].

The WMR cost functional has three weights, w_1 for the mobility, w_2 for hardware complexity, and w_3 for energy consumption. The weights are designed according to the importance of each aspect to the platform purpose. The sum of the three weights is always one. The proposed WMR cost function is:

$$J_r = w_1 \frac{C_m}{M_{ax_m}} + w_2 \frac{C_k}{M_{ax_k}} + w_3 \frac{C_e}{M_{ax_e}}, \tag{7.9}$$

where M_{ax_m}, M_{ax_k}, and M_{ax_e} are the maximum cost values for the linear velocities errors, hardware complexity, and energy consumption.

The simulation results of the previous examples are used to calculate the robots cost independently of each other. The following steps are uesd for each example. The first step is to study the cost functional performance if one of the aspects has less importance among the others as shown. The weight of the lowest important aspect is assigned with the value of 0.1, and the remaining aspects will have weights varying from $(0 - 0.9)$ taking into consideration that the sum of the two weights are always 0.9. The second step is to demonstrate the robots cost functionals with respect to one important aspect representing the overall cost values for different weightings $w = (w_1, w_2, w_3)$.

7.4.4.1 Driving in 3DOFs

For the 3DOFs example, the results of the first step are shown in Figure (7.9) The C3P gives the minimum cost value if the mobility aspect weight is reduced to the value of $w_1 = 0.1$ (Fig.7.9a), because it has the simplest hardware construction and consumes the least energy in comparison to the other robots. The importance of energy consumption appears in Figure(7.9b), with constant $w_2 = 0.1$ the C3P gives minimum cost value as long as the mobility weighting is $w_1 \leq 0.5$ and $w_3 \geq 0.35$, otherwise HCWR is the cheapest and then the C3P takes second.

By assigning the lowest importance to the energy consumption aspect, the C3P has the minimum cost, as long as the mobility weighting does not exceed 0.33 (Fig.7.9c).

The second step demonstrates the cost values in Table 7.2. If the hardware is the main important aspect in this comparison the weights is $w = (0.1, 0.8, 0.1)$ (as an example), then the minimum cost is found in Rasmis II and the C3P is the second cheapest robot. This is logical since the C3P has the largest velocity error and the simplest construction.

If the mobility complexity (for example $w = (0.8, 0.1, 0.1)$) is the most important aspect, then the HCWR platform is the cheapest one, while the C3P comes

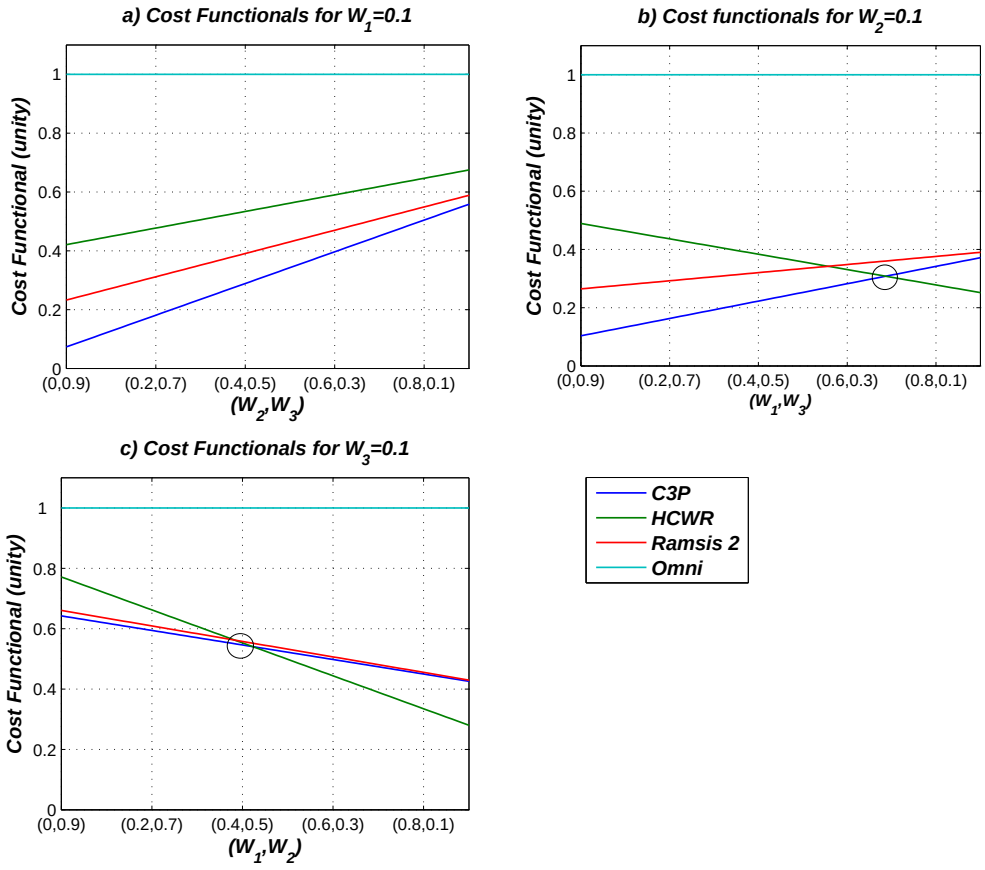

Figure 7.9. Evaluating the robots cost values for different weights for 3DOF

in third place. On the other hand , if the energy consumptions (for example $w = (0.1, 0.1, 0.8)$) is the most important aspect, then C3P has the minimum cost functional values and Ramsis II comes in second place.

The main point is that the weighting should be chosen according to the producer. In this work, the C3P hardware construction is chosen for its simplicity and its minimal consumption of energy; therefore the hardware complexity and energy consumption share the same importance as the mobility. As a result, the following weights are chosen: $w = (0.33, 0.33, 0.34)$ and corresponding to Table 7.3. The platform Ramsis II is found to be the cheapest robot, while the C3P comes in second place.

Table 7.2. Robots Cost Values for Driving in 3DOF

(W_1, W_2, W_3) RobotType	C3P	HCWR	ODWR	Ramsis II
(0.1, 0.8, 0.1)	***0.63***	0.7	1.00	0.62
(0.8, 0.1, 0.1)	***0.56***	0.23	1.00	0.40
(0.1, 0.1, 0.8)	***0.16***	0.45	1.00	0.28
(0.33, 0.33, 0.34)	***0.44***	0.46	1.00	0.43

7.4.4.2 Driving in (∞) Shape

The second example (driving in (∞) shape) will have different results and evaluations. The results illustrating the first step are presented in Figure (7.10).

The C3P gives the minimum cost value if the mobility aspect weight is reduced to the value of $w_1 = 0.1$ (Fig.7.10a), because it has the simplest hardware construction and consumes the least energy in comparison to the other robots. The importance of energy consumption appears in Figure(7.10b), with constant $w_2 = 0.1$ the C3P gives minimum cost value as long as the mobility weighting is $w_1 \leq 0.59$ and $w_3 \geq 0.31$, otherwise Ramsis II is the cheapest. The C3P takes second place as the cheapest robot, till it reaches $w_1 = 0.75$ and $w_3 = 0.15$, then the HCWR is the cheapest.

By assigning the lowest importance to the energy consumption aspect, the C3P has the lowest cost as long as the mobility weighting does not exceed 0.1 (Fig.7.10c).

Table. 7.3 represents the overall cost values for different weightings $w = (w_1, w_2, w_3)$.

If the hardware is the main important aspect in this comparison the weights will be $w = (0.1, 0.8, 0.1)$ (as an example), then the lowest cost will be found in the C3P. If the mobility complexity (for example $w = (0.8, 0.1, 0.1)$) is the most important aspect, then the HCWR platform is the cheapest one, while the C3P comes in third place. On the other hand, if the energy consumption (for example $w = (0.1, 0.1, 0.8)$) is the important aspect, then C3P has the minimum

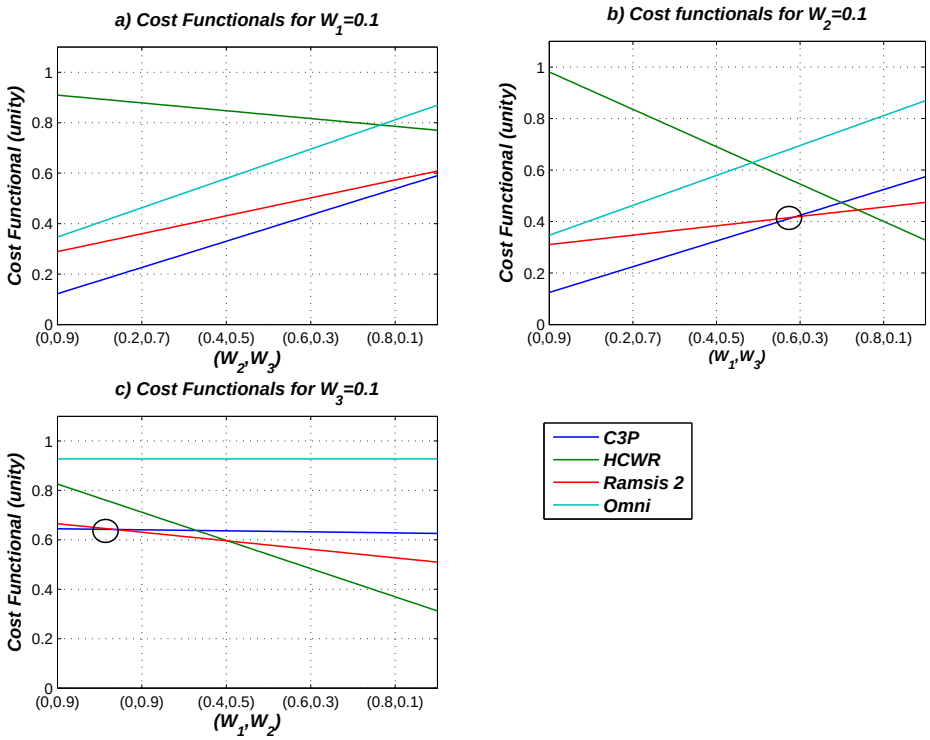

Figure 7.10. Evaluating the robots cost values for different weights for (∞) Shape

cost functional values and Ramsis II comes in second place. While by dividing the weights equally as chosen $w = (0.33, 0.33, 0.34)$,the C3P is the cheapest cost among the four platforms.

7.5 Summary

This chapter presented a comparison between four different holonomic mobile robots platforms: the C3P, the HCWR (Holonomic Caster Wheel Robot), the omnidirectional wheeled mobile robot, and Ramsis II. The comparison was done according to three main aspects: a) the mobility, b)the total energy consumed by each robot in a finite interval of time, and c) the hardware complexity. Inverse kinematic and forward dynamic models were presented for each robot for the simulation process. The simulation results illustrated the performance of each robot in comparison to the others. A cost functional was obtained to demonstrate the comparison, and a criteria was developed to measure the hardware complexity of

Table 7.3. Robots Cost Values for Driving in (∞) Shape

(W_1, W_2, W_3) \ RobotType	C3P	HCWR	ODWR	Ramsis II
(0.1, 0.8, 0.1)	***0.63***	0.75	0.92	0.65
(0.8, 0.1, 0.1)	***0.62***	0.25	0.92	0.49
(0.1, 0.1, 0.8)	***0.18***	0.98	0.42	0.33
(0.33, 0.33, 0.34)	***0.47***	0.63	0.75	0.49

each robot. The criteria was developed corresponding to the number and complexity of each actuator, sensor, wheel, and bearing unit. The electrical complexity was ignored due to its simple construction. The weighted sum method was used to find the cost value for each robot in respect of the importance of each aspect. In our work the simplicity of construction and lower energy consumption share the same importance as mobility. Based on that, the cost functional weights were assigned and the C3P was shown to have the most advantageous cost compared to the other holonomic mobile robots.

Chapter 8

Conclusions and Future Work

The main objective of this work is to deliver a holonomic wheeled mobile robot, which is simple in construction and efficient in performance. Therefore, a control scheme for a new holonomic wheeled mobile robot (C3P) is proposed in this thesis. The C3P has three caster wheels with wheel angular actuation, and sensed by the wheel angular velocities and steering angle values.

The C3P platform constraints were used to prove its holonomic characteristics based on the Frobenius Theorem. However, the actuation analysis revealed that the C3P has a singularity problem for some steering angles values. The problem is defined as the following: when all the wheel yield the same steering angle value the C3P cannot be actuated in the direction perpendicular to the wheel velocity vector. Therefore, a new actuated inverse kinematic solution was obtained to escape the singularity conditions. As a basic idea the coupling between any two wheel velocities is used to virtually actuate the steering angular velocity of the third wheel. The solution is termed the **W**heel **C**oupling **E**quation (WCE). The C3P's forward dynamic model using the Euler-Lagrangian method was obtained. The forward dynamic model has two equations of motion: the WTD (**W**heel **T**orques **D**ynamics) to calculate the wheel angular velocities with respect to the actuated wheels torques, and the DSE (**D**ynamic **S**teering **E**stimator) for the calculation of the steering angles and steering angular velocities corresponding to the angular wheel velocities and accelerations.

The kinematic motion control structure consists of two parts; a) the WCE regulator to overcome the singularity problem and adjust the steering angles to the

desired value, and b) regular PID controller to maintain the reference robot velocities with respect to the floor frame of coordinates. The simulation and practical outputs illustrated the acceptable performance of the velocity controller.

The inverse dynamic solution is developed using the Euler-Lagrangian method and using the inverse kinematic solution for actuating the steering and contact angular velocities within the solution. The solution is used in the development of the C3P velocity and position controllers (Chapter 4). The simulation and practical results yield less displacment errors in comparison to the inverse kinematics solution.

Finally, a comparison is made between the C3P and the other three holonomic mobile robots, which are defined as the following: **H**olonomic **C**aster **W**heeled **R**obot (HCWR), **O**mni **D**irectional **W**heeled **R**obot (ODWR), and RAMSIS II. The comparison is made in two steps: first, simulation comparison with respect to; the energy consumed by the robot to drive in a specific direction during a finite interval of time, the trajectory error of the robot, and the robot output velocities. Second, a cost function is assigned by using the weight sum method corresponding to three main aspects: a) the mobility, b)the total energy consumed by each robot in a finite interval of time, and c) the hardware complexity. Based on the simplicity of construction and lower energy consumption sharing the same importance as the mobility, the cost functional weights were assigned and the C3P was shown to have the most advantageous cost compared to the other holonomic mobile robots.

The C3P existing prototype has few disadvantages affecting its mobility performance. Due to the slippage and the friction, errors in the odometry calculations exist, leading to errors in trajectory and position update results. These factors generate disturbances that result in changing the steering angle configuration, which may lead to instabilities. For studying the system stability, a Lyapunov direct method is considered. The candidate Lyapunov function presented in this work is a quadratic one extracted from the robot kinetic energy equation. However, this function cannot be considered as a Lyapunov function because its first derivative has positive values for some steering angles configurations. The used Lyapunov function leads to two different regions, where the first derivative can be negative semi-definite or positive semi-definite. If the first derivative is negative semi-definite, a Lyapunov function exists and the system is always stable. On the

other hand, if the first derivative is positive semi-definite, no statement on stability can be made using the candidate function. To overcome these problems, the wheel should have a powerful torque controller and sensor for slippage detection, along with a mathematical algorithm for slippage modeling. The trajectory controller should be developed to use predesigned trajectories that keep the system in the negative semi-definite region of the used quadratic equation.

The future work can be defined in the following points: a) modeling the platform friction and wheel slippage, then integrating them within the dynamics problem, b) analysing the system non-linearities in more detail, c) developing a prediction algorithm to define the stability of the system according to the instantaneous states, d) developing a trajectory generating algorithm so the robot will avoid the non-defined stability regions using a predesigned trajectory.

Appendix A

Kinematics Modeling

A.1 The Velocity Generalized Wheel Jacobian

In [4] a method is developed to obtain the robot velocities solution described by the Jacobian for the generalized wheel. This jacobian is presented as follows

$$\begin{bmatrix} {}^{\bar{B}}\dot{x}_B \\ {}^{\bar{B}}\dot{y}_B \\ {}^{\bar{B}}\dot{\phi}_B \end{bmatrix} = \begin{bmatrix} \cos({}^B\theta_C) & -\sin({}^B\theta_C) & {}^Bd_{C_y} & -{}^Bd_{H_y} \\ \sin({}^B\theta_C) & \cos({}^B\theta_C) & -{}^Bd_{C_x} & {}^Bd_{H_x} \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} {}^{\bar{C}}\dot{x}_C \\ {}^{\bar{C}}\dot{y}_C \\ {}^{\bar{C}}\dot{\theta}_C \\ {}^H\dot{\theta}_S \end{bmatrix} \tag{A.1}$$

where

F *Floor*: The stationary reference co-ordinate system, where the z-axis is orthogonal to the planar surface.

B *Body* : The WMR body co-ordinate system.

H *Hip* : The co-ordinate system which moves with the body for the steering joint.

S *Steering* : The steering co-ordinate system which moves with the steering link with z-axis coincident with the z-axis of the Hip.

C *Contact Point* : The contact point co-ordinate system.

$\overline{B}$ *Instantaneously Coincident Body* : The co-ordinate system Coincident with the B co-ordinate system relative to the stationary F co-ordinate system.

$\overline{C}$ *Instantaneously Coincident Contact Point* : The co-ordinate system Coincident with the C co-ordinate system relative to the stationary F co-ordinate system.

The parameter ${}^{i}\theta_{j}$ is the angle between co-ordinates i and j, ${}^{i}d_{j_k}$ is the distance between co-ordinates i and j along k axes and ${}^{\bar{i}}\dot{k}_{j}$ is the linear velocity between co-ordinates i and j along k axes. Where $(i, j) \in (F, B, H, S, C, \overline{B}, \overline{C})$ and $k \in (x, y)$. Equation (A.1) is the pseudo-Jacobian matrix for the generalized wheel

$$ {}^{\bar{B}}\dot{\mathbf{p}}_{B} = \hat{J}\ \hat{\dot{\mathbf{q}}} \tag{A.2} $$

The wheel velocity vectors $\dot{\mathbf{q}}$ contains all the possible velocities which maybe actuated or sensed by the wheel, but a typical velocity vector contains fewer than four wheel variables. Since all the velocities of the wheel are rotational, the velocity vector should contain angular velocities of the wheel. As a result, $\hat{\dot{\mathbf{q}}} = W\ \dot{\mathbf{q}}$ where, W is the relation between the angular wheel velocities vector and the linear/angular wheel velocities vector, and

$$ {}^{\bar{B}}\dot{\mathbf{p}}_{B} = \hat{J}\ W\ \dot{\mathbf{q}} = J\ \dot{\mathbf{q}} \tag{A.3} $$

The rank of the wheel Jacobian matrix J indicates the number of DOFs of the wheel, where the wheel with fewer DOFs than the wheel variables is a redundant wheel.

A.1.1 The Acceleration Wheel Jacobian

The wheeled mobile robot accelerations are normally calculated by differentiating the robot velocities equation relation (A.1).

$$ \begin{bmatrix} {}^{\bar{B}}\ddot{x}_{B} \\ {}^{\bar{B}}\ddot{y}_{B} \\ {}^{\bar{B}}\ddot{\phi}_{B} \end{bmatrix} = \begin{bmatrix} \cos({}^{B}\theta_{C}) & -\sin({}^{B}\theta_{C}) & {}^{B}d_{C_y} & -{}^{B}d_{H_y} \\ \sin({}^{B}\theta_{C}) & \cos({}^{B}\theta_{C}) & -{}^{B}d_{C_x} & {}^{B}d_{H_x} \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} {}^{\bar{C}}\ddot{x}_{C} \\ {}^{\bar{C}}\ddot{y}_{C} \\ {}^{\bar{C}}\ddot{\theta}_{C} \\ {}^{H}\ddot{\theta}_{S} \end{bmatrix} $$

$$+\begin{bmatrix} {}^{B}d_{C_x} & {}^{B}d_{H_x} & {}^{B}d_{H_x} \\ {}^{B}d_{C_y} & {}^{B}d_{H_y} & {}^{B}d_{H_y} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} {}^{\bar{C}}\dot{\theta}_C^2 \\ -2\ {}^{\bar{C}}\dot{\theta}_C\ {}^{H}\dot{\theta}_S \\ {}^{H}\dot{\theta}_S^2 \end{bmatrix} \tag{A.4}$$

The robot accelerations vector ${}^{\bar{B}}\ddot{\mathbf{p}}_{B}$ is composed of three components: the self acceleration (${}^{\bar{C}}\ddot{X}_C$, ${}^{\bar{C}}\ \ddot{Y}_C$, ${}^{\bar{C}}\ddot{\theta}_C$ and ${}^{H}\ddot{\theta}_S$); the Centripetal accelerations (${}^{\bar{C}}\dot{\theta}_C^2$ and ${}^{H}\dot{\theta}_S^2$); and the Coriolis accelerations (${}^{\bar{C}}\dot{\theta}_C\ {}^{H}\dot{\theta}_S$).

A.1.2 Actuated Inverse and Sensed Forward Kinematics

The kinematic equations of motion for the wheels on the WMR will be combined in the composite robot equation, which is used in interpreting the WMR mobility, actuation and sensing properties.

$$A_o\ \dot{\mathbf{p}} = B_o\ \dot{\mathbf{q}} \tag{A.5}$$

The composite robot equation combines the Jacobian kinematic robot velocities for the "k" numbers of wheels in one Jacobian matrix $B_o(n \text{ x } n)$. The Matrix A_o is an identity matrix I with dimensions (n X 3), and they are used in the following composit equation.

$$\begin{bmatrix} I \\ I \\ . \\ . \\ I \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_1 & 0 & . & . & 0 \\ 0 & J_2 & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & . & . & J_k \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ . \\ . \\ \dot{\mathbf{q}}_k \end{bmatrix} \tag{A.6}$$

Investigating the actuation characteristics requires the distinction between the actuated and the non actuated variables, where $\dot{\mathbf{q}}_a$ and $\dot{\mathbf{q}}_n$ are the actuated and non actuated wheels velocities respectively, furthermore, J_{a_i} and J_{n_i} are the actuated and non-actuated Jacobians for wheel i. The actuated inverse solution is a relation between the robot velocities $\dot{\mathbf{p}}$ (as an input) and the actuated wheels velocities $\dot{\mathbf{q}}_a$(as and output). Its main function is to calculate a reference value fed to the actuated wheels velocities (motors) with respect to the desired robot input velocities. In order to control the robot velocities, a sensed forward solution

should be obtained and its main function to give the actual robot velocities with respect to the wheels sensed velocities $\dot{\mathbf{q}}_{sen}$.

The inverse solution is normally derived from the composite equation (A.6), where the actuated and non-actuated velocities are separated. The kinematic algorithms in [4] developed the following actuated inverse solution

$$\dot{\mathbf{q}}_a = \begin{bmatrix} \left[J_{a_1}^T \ \Delta\left(J_{n_1}\right) J_{a_1}\right]^{-1} J_{a_1}^T \Delta\left(J_{n_1}\right) \\ \left[J_{a_2}^T \ \Delta\left(J_{n_2}\right) J_{a_2}\right]^{-1} J_{a_2}^T \Delta\left(J_{n_2}\right) \\ \cdot \\ \cdot \\ \left[J_{a_k}^T \ \Delta\left(J_{n_k}\right) J_{a_k}\right]^{-1} J_{a_k}^T \Delta\left(J_{n_k}\right) \end{bmatrix} \dot{\mathbf{p}} = J_a \ \dot{\mathbf{p}} \tag{A.7}$$

The sensing characteristics is similar to the actuation characteristics, but the actuation inverse solution may not be the inversion of the sensed forward solution, because the sensed wheel velocities may not be the same actuated wheel velocities. The wheel velocities will be separated into $\dot{\mathbf{q}}_{sen}$ and $\dot{\mathbf{q}}_u$ sensed and non sensed wheels velocities, furthermore, J_{s_i} and J_{u_i} are the sensed and non-sensed Jacobian matrices for wheel i. The sensed forward solution is derived from the sensed composite robot equation to give the following solution

$$\dot{\mathbf{p}} = \left[\Delta\left(J_{u_1}\right) + \Delta\left(J_{u_2}\right) + ... + \Delta\left(J_{u_k}\right)\right]^{-1} \left[\Delta\left(J_{u_1}\right) J_{s_1} \ \ \Delta\left(J_{u_2}\right) J_{s_2} \ \ldots \ \Delta\left(J_{u_k}\right) J_{s_k}\right] \dot{\mathbf{q}}_{sen} \tag{A.8}$$

where the criterion $\Delta(U)$, which is used in equation A.7 and A.8, is described as

$$\Delta(U) = \{ \begin{matrix} -I & for\ U = 0 \\ U(U^T U)^{-1} U^T - I & otherwise \end{matrix} \tag{A.9}$$

Appendix B

The Dynamic Steering Estimator (DSE)

The Dynamic Steering Estimator torque equation is

$$\tau_s = M_{sx}\ddot{\mathbf{q}}_{sx} + G_{sx}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{B.1}$$

The inertia matrix M_{sx} can be divided into two matrices, M_x contains the mass and inertia parameters related to the wheels angular accelerations $\ddot{\mathbf{q}}_x$, the other matrix M_s contains the parameters related to the steering angular accelerations $\ddot{\mathbf{q}}_s$

$$\tau_s = \begin{bmatrix} M_x & M_s \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{bmatrix} + G_{sx}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{B.2}$$

where,

$$\tau_{s_{(3x1)}} = M_{x_{(3x3)}}\ \ddot{\mathbf{q}}_{x_{(3x1)}} + M_{s_{(3x3)}}\ \ddot{\mathbf{q}}_{s_{(3x1)}} + G_{sx_{(3x1)}}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{B.3}$$

Furthermore,

$$M_x = \frac{m_p\ h\ r}{9} \begin{bmatrix} C(\alpha_1 - \theta_{s_1}) & C(\alpha_1 - \theta_{s_2}) & C(\alpha_1 - \theta_{s_3}) \\ C(\alpha_2 - \theta_{s_1}) & C(\alpha_2 - \theta_{s_2}) & C(\alpha_2 - \theta_{s_3}) \\ C(\alpha_3 - \theta_{s_1}) & C(\alpha_3 - \theta_{s_2}) & C(\alpha_3 - \theta_{s_3}) \end{bmatrix} \tag{B.4}$$

and

$$M_s = \begin{bmatrix} I_p + m_p h^2 & I_p + m_p h^2\ C(\alpha_1 - \alpha_2) & I_p + m_p h^2\ C(\alpha_1 - \alpha_3) \\ I_p + m_p h^2\ C(\alpha_2 - \alpha_1) & I_p + m_p h^2 & I_p + m_p h^2\ C(\alpha_2 - \alpha_3) \\ I_p + m_p h^2\ C(\alpha_3 - \alpha_1) & I_p + m_p h^2\ C(\alpha_3 - \alpha_2) & I_p + m_p h^2 \end{bmatrix} /9 \tag{B.5}$$

The Centripetal and Coriolis forces are

$$G_{sx}(1) = \sum_{i=1}^{3} \sin(\alpha_i - \theta_{s_1})\dot{\theta}_{x_1}\dot{\theta}_{s_1} + h\sum_{i=1}^{3} \sin(\alpha_1 - \theta_{s_i})\dot{\theta}_{x_i}\dot{\theta}_{s_i}$$
$$+r\sum_{i=2}^{3} \sin(\theta_{s_1} - \theta_{s_i})\dot{\theta}_{x_1}\dot{\theta}_{x_i} \tag{B.6}$$

$$G_{sx}(2) = \sum_{i=1}^{3} \sin(\alpha_i - \theta_{s_2})\dot{\theta}_{x_2}\dot{\theta}_{s_1} + h\sum_{i=1}^{3} \sin(\alpha_2 - \theta_{s_i})\dot{\theta}_{x_i}\dot{\theta}_{s_i}$$
$$+r\sum_{i=1,3} \sin(\theta_{s_1} - \theta_{s_i})\dot{\theta}_{x_2}\dot{\theta}_{x_i} \tag{B.7}$$

$$G_{sx}(3) = \sum_{i=1}^{3} \sin(\alpha_i - \theta_{s_3})\dot{\theta}_{x_3}\dot{\theta}_{s_1} + h\sum_{i=1}^{3} \sin(\alpha_3 - \theta_{s_i})\dot{\theta}_{x_i}\dot{\theta}_{s_i}$$
$$+r\sum_{i=1}^{2} \sin(\theta_{s_i} - \theta_{s_3})\dot{\theta}_{x_3}\dot{\theta}_{x_i} \tag{B.8}$$

Appendix C

Inverse Dynamics Equations

C.1 The Inverse Dynamics Solution

By using Maple software the following torque equation is concluded

$$\tau_{x_a} = M_{sx_a}\ddot{\mathbf{q}}_{sx} + G_{sx_a}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s) \tag{C.1}$$

The matrix M_{sx_a} is separated in two (3X3) matrices shown below

$$M_{sx_a} = \left[\begin{array}{cc} M_{x_a} & M_{s_a} \end{array}\right], \quad \ddot{\mathbf{q}}_{sx} = \left[\begin{array}{c} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{array}\right] \tag{C.2}$$

where M_{xa} is a mass matrix constrained by the wheel angular acceleration vector $\ddot{\mathbf{q}}_x$

$$M_{x_a} = \left[\begin{array}{ccc} \frac{m_p}{9}r^2 & \frac{m_p r^2}{9}C(\theta_{s_1} - \theta_{s_2}) & \frac{m_p r^2}{9}C(\theta_{s_1} - \theta_{s_3}) \\ \frac{m_p r^2}{9}C(\theta_{s_2} - \theta_{s_1}) & \frac{m_p}{9}r^2 & \frac{m_p r^2}{9}C(\theta_{s_2} - \theta_{s_3}) \\ \frac{m_p r^2}{9}C(\theta_{s_3} - \theta_{s_1}) & \frac{m_p r^2}{9}C(\theta_{s_3} - \theta_{s_2}) & \frac{m_p}{9}r^2 \end{array}\right] \tag{C.3}$$

and M_{sa} is constrained by the steering angular acceleration vector $\ddot{\mathbf{q}}_s$

$$M_{s_a} = \left[\begin{array}{ccc} \frac{m_p rh}{9}\,C(\alpha_1 - \theta_{s_1}) & \frac{m_p rh}{9}\,C(\alpha_2 - \theta_{s_1}) & \frac{m_p rh}{9}\,C(\alpha_3 - \theta_{s_1}) \\ \frac{m_p rh}{9}\,C(\alpha_1 - \theta_{s_2}) & \frac{m_p rh}{9}\,C(\alpha_2 - \theta_{s_2}) & \frac{m_p rh}{9}\,C(\alpha_3 - \theta_{s_2}) \\ \frac{m_p rh}{9}\,C(\alpha_1 - \theta_{s_3}) & \frac{m_p rh}{9}\,C(\alpha_2 - \theta_{s_3}) & \frac{m_p rh}{9}\,C(\alpha_3 - \theta_{s_3} \end{array}\right], \tag{C.4}$$

where C stands for cos and S stands for sin. The Centripetal and Coriolis torques are presented by the vector G_{sx_a}, which has the following elements

$$G_{sx_a}(1) = \frac{m_p rh}{9}\, S(\alpha_1 - \theta_{s_1})\dot{\theta}_{s_1}^2 + \frac{m_p rh}{9}\, S(\alpha_2 - \theta_{s_1})\dot{\theta}_{s_1}\dot{\theta}_{s_2} + \frac{m_p rh}{9}\, S(\alpha_3 - \theta_{s_1})\dot{\theta}_{s_1}\dot{\theta}_{s_3}$$
$$-\frac{m_p r^2}{9}\, S(\theta_{s_1} - \theta_{s_2})\dot{\theta}_{s_1}\dot{\theta}_{x_2} - \frac{m_p r^2}{9}\, S(\theta_{s_1} - \theta_{s_3})\dot{\theta}_{s_1}\dot{\theta}_{x_3} + \frac{m_p r^2}{9}\, S(\theta_{s_1} - \theta_{s_2})\dot{\theta}_{s_2}\dot{\theta}_{x_2}$$
$$+\frac{m_p r^2}{9}\, S(\theta_{s_1} - \theta_{s_3})\dot{\theta}_{s_3}\dot{\theta}_{x_3}, \tag{C.5}$$

and

$$G_{sx_a}(2) = \frac{m_p rh}{9}\, S(\alpha_2 - \theta_{s_2})\dot{\theta}_{s_2}^2 + \frac{m_p rh}{9}\, S(\alpha_1 - \theta_{s_2})\dot{\theta}_{s_1}\dot{\theta}_{s_2} + \frac{m_p rh}{9}\, S(\alpha_3 - \theta_{s_2})\dot{\theta}_{s_2}\dot{\theta}_{s_3}$$
$$+\frac{m_p r^2}{9}\, S(\theta_{s_2} - \theta_{s_1})\dot{\theta}_{s_1}\dot{\theta}_{x_1} + \frac{m_p r^2}{9}\, S(\theta_{s_3} - \theta_{s_2})\dot{\theta}_{s_3}\dot{\theta}_{x_3} - \frac{m_p r^2}{9}\, S(\theta_{s_2} - \theta_{s_3})\dot{\theta}_{s_2}\dot{\theta}_{x_3}$$
$$-\frac{m_p r^2}{9}\, S(\theta_{s_2} - \theta_{s_1})\dot{\theta}_{s_2}\dot{\theta}_{x_1}, \tag{C.6}$$

and

$$G_{sx_a}(3) = \frac{m_p rh}{9}\, S(\alpha_3 - \theta_{s_3})\dot{\theta}_{s_3}^2 + \frac{m_p rh}{9}\, S(\alpha_1 - \theta_{s_3})\dot{\theta}_{s_1}\dot{\theta}_{s_3} + \frac{m_p rh}{9}\, S(\alpha_2 - \theta_{s_3})\dot{\theta}_{s_2}\dot{\theta}_{s_3}$$
$$+\frac{m_p r^2}{9}\, S(\theta_{s_3} - \theta_{s_1})\dot{\theta}_{s_1}\dot{\theta}_{x_1} + \frac{m_p r^2}{9}\, S(\theta_{s_3} - \theta_{s_2})\dot{\theta}_{s_2}\dot{\theta}_{x_2} - \frac{m_p r^2}{9}\, S(\theta_{s_3} - \theta_{s_2})\dot{\theta}_{s_3}\dot{\theta}_{x_2}$$
$$-\frac{m_p r^2}{9}\, S(\theta_{s_3} - \theta_{s_1})\dot{\theta}_{s_3}\dot{\theta}_{x_1}. \tag{C.7}$$

C.2 The Inverse Kinematics for Castor Wheel Acceleration Variables

Corresponding to the caster wheel closed chain described in chapter 2, the forward kinematic solution for the wheel i accelerations is

$$\ddot{\mathbf{p}} = J_{f_i}\ddot{\mathbf{q}}_i + J_{r_i}\ddot{\mathbf{q}}_{r_i} \tag{C.8}$$

where J_{f_i} is the forward direct solution for the wheel accelerations, and $J_{r_i}\dot{\mathbf{q}}_{r_i}$ is the Centripetal and Coriolis forces.

$$J_{f_i} = \begin{bmatrix} -r\sin(\theta_{s_i}) & h\cos(\alpha_i) + d\sin(\theta_{s_i}) & -h\cos(\alpha_i) \\ r\cos(\theta_{s_i}) & -h\sin(\alpha_i) + d\cos(\theta_{s_i}) & h\sin(\alpha_i) \\ 0 & 1 & -1 \end{bmatrix}, \ \dot{\mathbf{q}}_{r_i} = \begin{bmatrix} \dot{\theta}^2_{s_i} \\ -2\dot{\theta}_{c_i}\dot{\theta}_{s_i} \\ \dot{\theta}^2_{c_i} \end{bmatrix} \quad \text{(C.9)}$$

and

$$J_{r_i} = \begin{bmatrix} d_{sx_i} & d_{sx_i} & d_{x_i} \\ d_{sy_i} & d_{sy_i} & d_{y_i} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} h\sin(\alpha_i) & h\sin(\alpha_i) & h\sin(\alpha_i) - d\cos(\theta_{s_i}) \\ h\sin(\alpha_i) & h\cos(\alpha_i) & h\sin(\alpha_i) + d\cos(\theta_{s_i}) \\ 0 & 0 & 0 \end{bmatrix}. \quad \text{(C.10)}$$

In order to obtain the inverse kinematic solution for the C3P accelerations actuated variables, the procedure proposed in [4]. First the forward solution for the three caster wheels is combined in the following equation

$$A\ddot{\mathbf{p}} = B\ddot{\mathbf{q}} + D\ddot{\mathbf{q}}_r \quad \text{(C.11)}$$

where A is an identity matrix, B is the description of the forward solution for the three wheels together, and D is the Centripetal and Coriolis forces.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} J_{f_1} & 0 & 0 \\ 0 & J_{f_2} & 0 \\ 0 & 0 & J_{f_3} \end{bmatrix}, \quad D = \begin{bmatrix} J_{r_1} & 0 & 0 \\ 0 & J_{r_2} & 0 \\ 0 & 0 & J_{r_3} \end{bmatrix}. \quad \text{(C.12)}$$

Second, the wheel angular accelerations $\ddot{\mathbf{q}}_x$ and the steering angular accelerations $\ddot{\mathbf{q}}_s$, they are considered as the actuated variables and the non-actuated accelerations are $\ddot{\mathbf{q}}_c$. Therefore the matrix B will be separated into actuated vectors

(J_{a_i}) and un-actuated vectors (J_{n_i})

$$B = \begin{bmatrix} J_{a_1} & 0 & 0 & ||J_{n_1} & 0 & 0 \\ 0 & J_{a_2} & 0 & || \ 0 & J_{n_2} & 0 \\ 0 & 0 & J_{a_3} & || \ 0 & 0 & J_{n_3} \end{bmatrix}, \quad \ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \\ \ddot{\mathbf{q}}_c \end{bmatrix}. \tag{C.13}$$

After rearranging the Centripetal and Coriolis variables described by the following D matrix

$$D = \begin{bmatrix} d_{sx_1} & 0 & 0 & ||d_{x_1} & 0 & 0 & ||d_{sx_1} & 0 & 0 \\ d_{sy_1} & 0 & 0 & ||d_{y_1} & 0 & 0 & ||d_{sy_1} & 0 & 0 \\ \underline{0} & \underline{0} & \underline{0} & || \ \underline{0} & \underline{0} & \underline{0} & || \ \underline{0} & \underline{0} & \underline{0} \\ 0 & d_{sx_2} & 0 & || \ 0 & d_{x_2} & 0 & || \ 0 & d_{sx_2} & 0 \\ 0 & d_{sy_2} & 0 & || \ 0 & d_{y_2} & 0 & || \ 0 & d_{sy_2} & 0 \\ \underline{0} & \underline{0} & \underline{0} & || \ \underline{0} & \underline{0} & \underline{0} & || \ \underline{0} & \underline{0} & \underline{0} \\ 0 & 0 & d_{sx_3} & || \ 0 & 0 & d_{x_3} & || \ 0 & 0 & d_{sx_3} \\ 0 & 0 & d_{sy_3} & || \ 0 & 0 & d_{y_3} & || \ 0 & 0 & d_{sy_3} \\ 0 & 0 & 0 & || \ 0 & 0 & 0 & || \ 0 & 0 & 0 \end{bmatrix}, \quad \ddot{\mathbf{q}}_r = \begin{bmatrix} \dot{\theta}^2_{s_1} \\ \dot{\theta}^2_{s_2} \\ \dot{\theta}^2_{s_3} \\ \dot{\theta}^2_{c_1} \\ \dot{\theta}^2_{c_2} \\ \dot{\theta}^2_{c_3} \\ -2\dot{\theta}_{c_1}\dot{\theta}_{s_1} \\ -2\dot{\theta}_{c_2}\dot{\theta}_{s_2} \\ -2\dot{\theta}_{c_3}\dot{\theta}_{s_3} \end{bmatrix} \tag{C.14}$$

The accelerations inverse solution is obtained from (C.11)

$$B\ddot{\mathbf{q}} = A\ddot{\mathbf{p}} - D\dot{\mathbf{q}}_r \tag{C.15}$$

as the following

$$\ddot{\mathbf{q}} = B_{_T} A\ddot{\mathbf{p}} - B_{_T} D\dot{\mathbf{q}}_r, \quad where \quad B_{_T} = (B^T B)^{-1} B^T \tag{C.16}$$

then the actuated accelerations are concluded from the equation (C.16) as described

$$\begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{bmatrix}_{(6x1)} = B_{_{TA_{(6x9)}}} A_{(9x3)} \ddot{\mathbf{p}}_{(3x1)} - B_{_{TA(6x9)}} D_{(9x9)} \dot{\mathbf{q}}_{r_{(9x1)}} \tag{C.17}$$

The matrix $B_{_{TA}}$ is a part of the matrix $B_{_T}$, where $B_{_{TA}}$ contains the effective parts to actuate the accelerations $\ddot{\mathbf{q}}_x$ and $\ddot{\mathbf{q}}_s$. The matrix $B_{_{TA}}$ is a diagonalized matrix

with the dimensions of $(6x9)$

$$B_{TA} = \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix}, \tag{C.18}$$

where,

$$b_i = X_{r_i} J_{a_i}^T + \Lambda_i J_{n_i}^T \tag{C.19}$$

and,

$$X_{r_i} = -\left[J_{a_i}^T \Delta\left(J_{n_i}^T\right) J_{a_i}\right]^{-1} \tag{C.20}$$

$$\Lambda_i = -X_{r_i} J_{a_i}^T J_{n_i} \left(J_{n_i}^T J_{n_i}\right)^{-1} \tag{C.21}$$

$$\Delta\left(J_{n_i}^T\right) = J_{n_i} \left(J_{n_i}^T J_{n_i}\right)^{-1} J_{n_i}^T - I. \tag{C.22}$$

Finally, the inverse actuated kinematic accelerations are

$$\begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{bmatrix} = J_{in_v} \ddot{\mathbf{p}} - g_{cs}(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_c) \tag{C.23}$$

where

$$J_{in_v} = \begin{bmatrix} J_{in_x} \\ J_{in_s} \end{bmatrix} \tag{C.24}$$

By the aid of Maple, the vector g_{cs} is

$$\begin{aligned} g_{cs}(1) = \frac{1}{rd}(&-cos(\theta_{s_1})^2 \dot{x}^2 - \dot{y}^2 + \dot{y}^2 cos(\theta_{s_1})^2 - 2\, cos(\theta_{s_1}) sin(\theta_{s_1}) \dot{x}\dot{y} \\ &-2cos(\theta_{s_1}) sin(\theta_{s_1} - \alpha_1) \dot{x}\dot{\phi} - 2sin(\theta_{s_1}) sin(\theta_{s_1} - \alpha_1) \dot{y}\dot{\phi} - \dot{\phi}^2 h^2 \\ &+\dot{\phi}^2 h^2 cos(\theta_{s_1} - \alpha_1) - \dot{\phi}^2 d\, sin(\theta_{s_1} - \alpha_1)) \end{aligned} \tag{C.25}$$

$$g_{cs}(2) = \frac{-1}{d}(\dot{\phi}^2 h\, sin(\theta_{s_1} - \alpha_1)) \tag{C.26}$$

$$g_{cs}(3) = \frac{1}{rd}(-cos(\theta_{s_2})^2\dot{x}^2 - \dot{y}^2 + \dot{y}^2 cos(\theta_{s_2})^2 - 2\ cos(\theta_{s_2})sin(\theta_{s_2})\dot{x}\dot{y}$$

$$-2cos(\theta_{s_2})sin(\theta_{s_2} - \alpha_2)\dot{x}\dot{\phi} - 2sin(\theta_{s_2})sin(\theta_{s_2} - \alpha_2)\dot{y}\dot{\phi} - \dot{\phi}^2 h^2$$

$$+\dot{\phi}^2 h^2 cos(\theta_{s_2} - \alpha_2) - \dot{\phi}^2 d\ sin(\theta_{s_2} - \alpha_2)) \tag{C.27}$$

$$g_{cs}(4) = \frac{-1}{d}(\dot{\phi}^2 h\ sin(\theta_{s_2} - \alpha_2)) \tag{C.28}$$

$$g_{cs}(5) = \frac{1}{rd}(-cos(\theta_{s_3})^2\dot{x}^2 - \dot{y}^2 + \dot{y}^2 cos(\theta_{s_3})^2 - 2\ cos(\theta_{s_3})sin(\theta_{s_3})\dot{x}\dot{y}$$

$$-2cos(\theta_{s_3})sin(\theta_{s_3} - \alpha_3)\dot{x}\dot{\phi} - 2sin(\theta_{s_3})sin(\theta_{s_3} - \alpha_3)\dot{y}\dot{\phi} - \dot{\phi}^2 h^2$$

$$+\dot{\phi}^2 h^2 cos(\theta_{s_3} - \alpha_3) - \dot{\phi}^2 d\ sin(\theta_{s_3} - \alpha_3)) \tag{C.29}$$

$$g_{cs}(6) = \frac{-1}{d}(\dot{\phi}^2 h\ sin(\theta_{s_3} - \alpha_3)) \tag{C.30}$$

Appendix D

The Lyapunov Analysis

$$axx := \frac{1}{2} mp \left(\left(\frac{\sin(\theta_1)^2}{3 r^2 mp} + \frac{\sin(\theta_2)^2}{3 r^2 mp} + \frac{\sin(\theta_3)^2}{3 r^2 mp} \right)^2 + \left(- \frac{\cos(\theta_1) \sin(\theta_1)}{3 r^2 mp} - \frac{\cos(\theta_2) \sin(\theta_2)}{3 r^2 mp} - \frac{\cos(\theta_3) \sin(\theta_3)}{3 r^2 mp} \right)^2 \right) + \frac{2 I_p \left(- \frac{\cos(\alpha 1 - \theta_1) \sin(\theta_1)}{r^3 m_p} - \frac{\cos(\alpha 2 - \theta_2) \sin(\theta_2)}{r^3 m_p} - \frac{\cos(\alpha 3 - \theta_3) \sin(\theta_3)}{r^3 m_p} \right)^2}{(\cos(2 \alpha 1 - 2 \theta_1) + \cos(2 \alpha 2 - 2 \theta_2) + \cos(2 \alpha 3 - 2 \theta_3) + 3)^2}$$

$$ayy := \frac{1}{2} mp \left(\left(- \frac{\cos(\theta_1) \sin(\theta_1)}{3 r^2 mp} - \frac{\cos(\theta_2) \sin(\theta_2)}{3 r^2 mp} - \frac{\cos(\theta_3) \sin(\theta_3)}{3 r^2 mp} \right)^2 + \left(\frac{\cos(\theta_1)^2}{3 r^2 mp} + \frac{\cos(\theta_2)^2}{3 r^2 mp} + \frac{\cos(\theta_3)^2}{3 r^2 mp} \right)^2 \right) + \frac{2 I_p \left(\frac{\cos(\alpha 1 - \theta_1) \cos(\theta_1)}{r^3 m_p} + \frac{\cos(\alpha 2 - \theta_2) \cos(\theta_2)}{r^3 m_p} + \frac{\cos(\alpha 3 - \theta_3) \cos(\theta_3)}{r^3 m_p} \right)^2}{(\cos(2 \alpha 1 - 2 \theta_1) + \cos(2 \alpha 2 - 2 \theta_2) + \cos(2 \alpha 3 - 2 \theta_3) + 3)^2}$$

$$azz := \frac{1}{2} mp \left(\left(- \frac{\sin(\theta_1) h (\sin(\alpha 1) \sin(\theta_1) + \cos(\theta_1) \cos(\alpha 1))}{3 r^2 mp} - \frac{\sin(\theta_2) h (\sin(\alpha 2) \sin(\theta_2) + \cos(\theta_2) \cos(\alpha 2))}{3 r^2 mp} - \frac{\sin(\theta_3) h (\sin(\alpha 3) \sin(\theta_3) + \cos(\theta_3) \cos(\alpha 3))}{3 r^2 m_p} \right) ^2 + \left(\frac{\cos(\theta_1) h (\sin(\alpha 1) \sin(\theta_1) + \cos(\theta_1) \cos(\alpha 1))}{3 r^2 m_p} + \frac{\cos(\theta_2) h (\sin(\alpha 2) \sin(\theta_2) + \cos(\theta_2) \cos(\alpha 2))}{3 r^2 m_p} + \frac{\cos(\theta_3) h (\sin(\alpha 3) \sin(\theta_3) + \cos(\theta_3) \cos(\alpha 3))}{3 r^2 m_p} \right) ^2 \right) + \frac{1}{(\cos(2 \alpha 1 - 2 \theta_1) + \cos(2 \alpha 2 - 2 \theta_2) + \cos(2 \alpha 3 - 2 \theta_3) + 3)^2} \left(2 I_p \left(\frac{\cos(\alpha 1 - \theta_1) h (\sin(\alpha 1) \sin(\theta_1) + \cos(\theta_1) \cos(\alpha 1))}{r^3 m_p} + \frac{\cos(\alpha 2 - \theta_2) h (\sin(\alpha 2) \sin(\theta_2) + \cos(\theta_2) \cos(\alpha 2))}{r^3 m_p} + \frac{\cos(\alpha 3 - \theta_3) h (\sin(\alpha 3) \sin(\theta_3) + \cos(\theta_3) \cos(\alpha 3))}{r^3 m_p} \right) ^2 \right)$$

$$axy := \frac{1}{2} mp \left(2 \left(- \frac{\cos(\theta 1) \sin(\theta 1)}{3 r^2 mp} - \frac{\cos(\theta 2) \sin(\theta 2)}{3 r^2 mp} - \frac{\cos(\theta 3) \sin(\theta 3)}{3 r^2 mp} \right) \left(\frac{\sin(\theta 1)^2}{3 r^2 mp} + \frac{\sin(\theta 2)^2}{3 r^2 mp} + \frac{\sin(\theta 3)^2}{3 r^2 mp} \right) \right.$$

$$+ 2 \left(\frac{\cos(\theta_1)^2}{3 r^2 m_p} + \frac{\cos(\theta_2)^2}{3 r^2 m_p} + \frac{\cos(\theta_3)^2}{3 r^2 m_p} \right) \left(- \frac{\cos(\theta_1) \sin(\theta_1)}{3 r^2 m_p} - \frac{\cos(\theta_2) \sin(\theta_2)}{3 r^2 m_p} - \frac{\cos(\theta_3) \sin(\theta_3)}{3 r^2 m_p} \right) \Bigg) +$$

$$\frac{1}{(\cos(2\,\alpha 1 - 2\,\theta_1) + \cos(2\,\alpha 2 - 2\,\theta_2) + \cos(2\,\alpha 3 - 2\,\theta_3) + 3)^2} \left(4 I_p \left(\frac{\cos(\alpha 1 - \theta_1) \cos(\theta_1)}{r^3 m_p} \right. \right.$$

$$\left. + \frac{\cos(\alpha 2 - \theta_2) \cos(\theta_2)}{r^3 m_p} + \frac{\cos(\alpha 3 - \theta_3) \cos(\theta_3)}{r^3 m_p} \right) \left(- \frac{\cos(\alpha 1 - \theta_1) \sin(\theta_1)}{r^3 m_p} - \frac{\cos(\alpha 2 - \theta_2) \sin(\theta_2)}{r^3 m_p} \right.$$

$$\left. \left. - \frac{\cos(\alpha 3 - \theta_3) \sin(\theta_3)}{r^3 m_p} \right) \right)$$

$$axphi := \frac{1}{2} mp \left(2 \left(- \frac{\sin(\theta 1)\, h\, (\sin(\alpha 1) \sin(\theta 1) + \cos(\theta 1) \cos(\alpha 1))}{3 r^2 mp} \right. \right.$$

$$\left. - \frac{\sin(\theta_2)\, h\, (\sin(\alpha 2) \sin(\theta_2) + \cos(\theta_2) \cos(\alpha 2))}{3 r^2 m_p} - \frac{\sin(\theta_3)\, h\, (\sin(\alpha 3) \sin(\theta_3) + \cos(\theta_3) \cos(\alpha 3))}{3 r^2 m_p} \right) \left(\right.$$

$$\left. \frac{\sin(\theta_1)^2}{3 r^2 m_p} + \frac{\sin(\theta_2)^2}{3 r^2 m_p} + \frac{\sin(\theta_3)^2}{3 r^2 m_p} \right) + 2 \left(\frac{\cos(\theta_1)\, h\, (\sin(\alpha 1) \sin(\theta_1) + \cos(\theta_1) \cos(\alpha 1))}{3 r^2 m_p} \right.$$

$$\left. + \frac{\cos(\theta_2)\, h\, (\sin(\alpha 2) \sin(\theta_2) + \cos(\theta_2) \cos(\alpha 2))}{3 r^2 m_p} + \frac{\cos(\theta_3)\, h\, (\sin(\alpha 3) \sin(\theta_3) + \cos(\theta_3) \cos(\alpha 3))}{3 r^2 m_p} \right) \left(\right.$$

$$\left. - \frac{\cos(\theta_1) \sin(\theta_1)}{3 r^2 m_p} - \frac{\cos(\theta_2) \sin(\theta_2)}{3 r^2 m_p} - \frac{\cos(\theta_3) \sin(\theta_3)}{3 r^2 m_p} \right) \Bigg) +$$

$$Xdot(t) := K_x (X_g - X_i(t)) \left(1 - e^{(-|X_g - X_i(t)|)} \right)$$

$$Ydot(t) := K_y (Y_g - Y_i(t)) \left(1 - e^{(-|Y_g - Y_i(t)|)} \right)$$

$$Phdot(t) := K_{Ph} (Ph_g - Ph_i(t)) \left(1 - e^{(-|Ph_g - Ph_i(t)|)} \right) + K\_er \, (\arctan(-Y_g, X_g) - \arctan(-Y_g + Y_i(t), X_g - X_i(t)))$$

$$Xdotc := -K_x \left(\frac{d}{dt} X_i(t) \right) \left(1 - e^{(-|X_g - X_i(t)|)} \right) - K_x (X_g - X_i(t)) \, \mathrm{abs}(1, X_g - X_i(t)) \left(\frac{d}{dt} X_i(t) \right) e^{(-|X_g - X_i(t)|)}$$

$$Ydotc := -K_y \left(\frac{d}{dt} Y_i(t) \right) \left(1 - e^{(-|Y_g - Y_i(t)|)} \right) - K_y (Y_g - Y_i(t)) \, \mathrm{abs}(1, Y_g - Y_i(t)) \left(\frac{d}{dt} Y_i(t) \right) e^{(-|Y_g - Y_i(t)|)}$$

$$Phdotc := -K_{Ph} \left(\frac{d}{dt} Ph_i(t) \right) \left(1 - e^{(-|Ph_g - Ph_i(t)|)} \right) - K_{Ph} (Ph_g - Ph_i(t)) \, \mathrm{abs}(1, Ph_g - Ph_i(t)) \left(\frac{d}{dt} Ph_i(t) \right) e^{(-|Ph_g - Ph_i(t)|)} - \frac{K\_er \left(\frac{\frac{d}{dt} Y_i(t)}{X_g - X_i(t)} + \frac{(-Y_g + Y_i(t)) \left(\frac{d}{dt} X_i(t) \right)}{(X_g - X_i(t))^2} \right)}{1 + \frac{(-Y_g + Y_i(t))^2}{(X_g - X_i(t))^2}}$$

$$\left(\frac{\cos(\alpha 1 - \theta_1) \, h \, (\sin(\alpha 1) \sin(\theta_1) + \cos(\theta_1) \cos(\alpha 1))}{r^3 \, m_P} + \frac{\cos(\alpha 2 - \theta_2) \, h \, (\sin(\alpha 2) \sin(\theta_2) + \cos(\theta_2) \cos(\alpha 2))}{r^3 \, m_P} + \frac{\cos(\alpha 3 - \theta_3) \, h \, (\sin(\alpha 3) \sin(\theta_3) + \cos(\theta_3) \cos(\alpha 3))}{r^3 \, m_P} \right) \left(\frac{\cos(\alpha 1 - \theta_1) \cos(\theta_1)}{r^3 \, m_P} + \frac{\cos(\alpha 2 - \theta_2) \cos(\theta_2)}{r^3 \, m_P} + \frac{\cos(\alpha 3 - \theta_3) \cos(\theta_3)}{r^3 \, m_P} \right)$$

$$Xdot(t) := K_x\,(X_g - X_i(t))\left(1 - e^{\left(-\left|X_g - X_i(t)\right|\right)}\right)$$

$$Ydot(t) := K_y\,(Y_g - Y_i(t))\left(1 - e^{\left(-\left|Y_g - Y_i(t)\right|\right)}\right)$$

$$Phdot(t) := K_{Ph}\,(Ph_g - Ph_i(t))\left(1 - e^{\left(-\left|Ph_g - Ph_i(t)\right|\right)}\right) + K\_er\,(\arctan(-Y_g, X_g) - \arctan(-Y_g + Y_i(t), X_g - X_i(t)))$$

$$Bv := \left[\begin{array}{c} \left[-K_x\left(1 - e^{\left(-\left|X_g - X_i(t)\right|\right)}\right) - K_x\,(X_g - X_i(t))\,\mathrm{abs}(1, X_g - X_i(t))\,e^{\left(-\left|X_g - X_i(t)\right|\right)}, 0, 0\right], \\ \left[0, -K_y\left(1 - e^{\left(-\left|Y_g - Y_i(t)\right|\right)}\right) - K_y\,(Y_g - Y_i(t))\,\mathrm{abs}(1, Y_g - Y_i(t))\,e^{\left(-\left|Y_g - Y_i(t)\right|\right)}, 0\right], \\ \left[\begin{array}{c} -\dfrac{K\_er\,(-Y_g + Y_i(t))}{(X_g - X_i(t))^2\left(1 + \dfrac{(-Y_g + Y_i(t))^2}{(X_g - X_i(t))^2}\right)}, -\dfrac{K\_er}{(X_g - X_i(t))\left(1 + \dfrac{(-Y_g + Y_i(t))^2}{(X_g - X_i(t))^2}\right)}, \\ -K_{Ph}\left(1 - e^{\left(-\left|Ph_g - Ph_i(t)\right|\right)}\right) - K_{Ph}\,(Ph_g - Ph_i(t))\,\mathrm{abs}(1, Ph_g - Ph_i(t))\,e^{\left(-\left|Ph_g - Ph_i(t)\right|\right)} \end{array}\right] \end{array}\right]$$

Appendix E

The Kinematics and Dynamics Modeling of Different Holonomic Wheeled Mobile Robot

E.1 Kinematics Modeling of Holonomic Mobile Robots

In this section, the method proposed in [4] to obtain the inverse and the forward kinemaitcs models will be used. The method is explained in detailes in Appendix A

E.1.1 Holonomic Caster Wheeled Robot (HCWR)

Corresponding to the HCWR configuration descriped in chapter 7 with Fig. (7.1), the actuated ($\dot{\mathbf{q}}_a$) and un-actuated wheels ($\dot{\mathbf{q}}_n$) velocities vectors are

$$\dot{\mathbf{q}}_a = \begin{bmatrix} \dot{\mathbf{q}}_x \\ \dot{\mathbf{q}}_s \end{bmatrix} = \begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \\ \dot{\theta}_{s_1} \\ \dot{\theta}_{s_2} \\ \dot{\theta}_{s_3} \end{bmatrix}, \quad \dot{\mathbf{q}}_n = \dot{\mathbf{q}}_c = \begin{bmatrix} \dot{\theta}_{c_1} \\ \dot{\theta}_{c_2} \\ \dot{\theta}_{c_3} \end{bmatrix}, \tag{E.1}$$

as a result the actuated (J_{a_i}) and non actuated wheel (J_{n_i}) Jacobians for the i^{th} wheel are

$$J_{a_i} = \begin{bmatrix} -r\sin(\theta_{s_i}) & -h\cos(\alpha_i) \\ r\cos(\theta_{s_i}) & h\sin(\alpha_i) \\ 0 & -1 \end{bmatrix}, \quad J_{n_i} = \begin{bmatrix} h\cos(\alpha_i) + d\sin(\theta_{s_i}) \\ -h\sin(\alpha_i) + d\cos(\theta_{s_i}) \\ 1 \end{bmatrix}. \tag{E.2}$$

The substitution of the actuated and un-actuated Jacobian (E.2) in the inverse kinematics genralized solution in equation (A.7) the actuated inverse will be

$$\dot{\mathbf{q}}_a = J_{inv}\ \dot{\mathbf{p}}, \tag{E.3}$$

$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \\ \dot{\theta}_{s_1} \\ \dot{\theta}_{s_2} \\ \dot{\theta}_{s_3} \end{bmatrix} = \begin{bmatrix} \frac{-\sin(\theta_{s_1})}{r} & \frac{\cos(\theta_{s_1})}{r} & \frac{h\cos(\alpha_1-\theta_{s_1})}{r} \\ \frac{-\sin(\theta_{s_2})}{r} & \frac{\cos(\theta_{s_2})}{r} & \frac{h\cos(\alpha_2-\theta_{s_2})}{r} \\ \frac{-\sin(\theta_{s_3})}{r} & \frac{\cos(\theta_{s_3})}{r} & \frac{h\cos(\alpha_3-\theta_{s_3})}{r} \\ \frac{-\cos(\theta_{s_1})}{d} & \frac{-\sin(\theta_{s_1})}{d} & \frac{h\sin(\alpha_1-\theta_{s_1})-d}{d} \\ \frac{-\cos(\theta_{s_2})}{d} & \frac{-\sin(\theta_{s_2})}{d} & \frac{h\sin(\alpha_2-\theta_{s_2})-d}{d} \\ \frac{-\cos(\theta_{s_3})}{d} & \frac{-\sin(\theta_{s_3})}{d} & \frac{h\sin(\alpha_3-\theta_{s_3})-d}{d} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}, \tag{E.4}$$

which provides non singular matrix (J_{inv}) for any steering angles configuration. The sensing forward solution of the HCWR is exactly the same as the solution obtained for the C3P, since they have the same sensing elements.

E.1.2 Omni Directional Wheeled Robot (ODWR)

The omni-directional wheels have become popular in WMRs field due to their holonomic mobility features. The Omni-directional wheel is based on the following concept; while the wheel drive in the direction normal to the motor axis, the wheel can slide frictionless in the motor axis direction [110] [111] . The omni-directional wheel jachobian is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -R\sin(\theta_i) & r_o\sin(\theta_i+\eta_i) & d_{y_i} \\ R\cos(\theta_i) & -r_o\cos(\theta_i+\eta_i) & -d_{x_i} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{x_i} \\ \dot{\theta}_{p_i} \\ \dot{\theta}_{c_i} \end{bmatrix} \quad for\ i \in \{1,2,3\} \tag{E.5}$$

where $\dot{\theta}_{x_i}$ is the wheel angular velocity, $\dot{\theta}_{p_i}$ is the roller angular velocity around the z axis, and $\dot{\theta}_{c_i}$ the contact point angular velocity.

The Omni-directioanl Wheeled Robot (ODWR) used in this thesis is shown in Fig. 7.2, where the geometric parameters of the robot are set to be as close as possible to the C3P geometric dimensions; r= 0.01 m (the radius of the roller), R=0.08 m (the radius of the wheel), and h= 0.343 m (distance between platform co-ordinates of reference and each contact point). The angle shifted between each wheel is symmetric to 120^o as shown in the figure. Hence, the value of $\theta_{_1} = 90^o$, $\theta_{_1} = 210^o$, and $\theta_{_3} = 300^o$ while $\eta_{_i} = 90^o$ for $i \in \{1, 2, 3\}$. As a result, the Jacobian of each wheel with respect to the robot instantaneous frame of co-ordinates are

$$\dot{\mathbf{p}} = \begin{bmatrix} -R & 0 & h \\ 0 & r_o & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{p_1} \\ \dot{\theta}_{c_1} \end{bmatrix}, \tag{E.6}$$

$$\dot{\mathbf{p}} = \begin{bmatrix} -R/2 & \sqrt{3}r_o/2 & -h/2 \\ \sqrt{3}R/2 & -r_o/2 & -\sqrt{3}h/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{x_2} \\ \dot{\theta}_{p_2} \\ \dot{\theta}_{c_2} \end{bmatrix}, \tag{E.7}$$

$$\dot{\mathbf{p}} = \begin{bmatrix} R/2 & -\sqrt{3}r_o/2 & -h/2 \\ -\sqrt{3}R/2 & -r_o/2 & \sqrt{3}h/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{x_3} \\ \dot{\theta}_{p_3} \\ \dot{\theta}_{c_3} \end{bmatrix}. \tag{E.8}$$

The angular wheel velocities $\dot{\theta}_{x_i}$ for $i \in \{1, 2, 3\}$ are the actuated elements. Hence, the actuated ($\dot{\mathbf{q}}_a$) and un-actuated ($\dot{\mathbf{q}}_n$) wheels velocities vectors are

$$\dot{\mathbf{q}}_a = \dot{\mathbf{q}}_x = \begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix}, \quad \dot{\mathbf{q}}_n = \begin{bmatrix} \dot{\mathbf{q}}_p \\ \dot{\mathbf{q}}_c \end{bmatrix} = \begin{bmatrix} \dot{\theta}_{p_1} \\ \dot{\theta}_{p_2} \\ \dot{\theta}_{p_3} \\ \dot{\theta}_{c_1} \\ \dot{\theta}_{c_2} \\ \dot{\theta}_{c_3} \end{bmatrix} \tag{E.9}$$

which result the following actuated and non actuated i^{th} wheel Jacobians

$$\dot{J}_a = \begin{bmatrix} -R\sin(\theta_i) \\ -R\cos(\theta_i) \\ 0 \end{bmatrix}, \quad \dot{J}_n = \begin{bmatrix} r\sin(\theta_i+\eta_i) & dy_i \\ r\cos(\theta_i+\eta_i) & -dx_i \\ 0 & 1 \end{bmatrix} \tag{E.10}$$

After substituting equation (E.10) in the inverse kinematics genralized solution in equation (A.7) the inverse kinematic solution will be

$$\begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} -1 & 0 & h \\ 1/2 & \sqrt{3}/2 & h \\ 1/2 & -\sqrt{3}/2 & h \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}, \tag{E.11}$$

which yields non-singular matrix and gives actuated 3DOF's. The sensing elements are the angular wheeled velocities as well. Thus the forward kinematic solution will deliver a robust values for the robot velocities through the following relation

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = R \begin{bmatrix} -2/3 & 1/3 & 1/3 \\ 0 & 1/\sqrt{3} & -1/\sqrt{3} \\ 1/(3h) & 1/(3h) & 1/(3h) \end{bmatrix} \begin{bmatrix} \dot{\theta}_{x_1} \\ \dot{\theta}_{x_2} \\ \dot{\theta}_{x_3} \end{bmatrix}. \tag{E.12}$$

E.1.3 Ramsis II

Ramsis II is the name of the mobile robot developed at the Automatic Control Laboratory of the Swiss Federal institute of Technology (ETH) in Zurich [102]. The basic configuration of the mobile platform is shown in Fig. 7.3. The turret part is presented in red color (Fig. 7.3) with offset b, which provides the third DOF, and the composite equation of the robot is

$$A_o\ \dot{\mathbf{p}} = B_{op}\ \dot{\mathbf{q}} \tag{E.13}$$

$$A_o\dot{\mathbf{p}} = \begin{bmatrix} -rS(\theta_s) & -bC(\theta_s)+aS(\theta_s) & 0 & 0 & 0 \\ rC(\theta_s) & -bS(\theta_s)-aC(\theta_s) & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -rS(\theta_s) & -bC(\theta_s)-aS(\theta_s) \\ 0 & 0 & 0 & rC(\theta_s) & -bS(\theta_s)+aC(\theta_s) \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{xr_1} \\ \dot{\theta}_{zr_1} \\ \dot{\theta}_s \\ \dot{\theta}_{xr_2} \\ \dot{\theta}_{zr_2} \end{bmatrix} \tag{E.14}$$

where $\dot{\theta}_{xr_i}$ is the wheel angular velocity and $\dot{\theta}_{zr_i}$ is the contact angular velocity $i \in \{1, 2\}$, while $\dot{\theta}_s$ is the turret angular velocity. The actuated velocities ($\dot{\mathbf{q}}_a$) and non actuated velocities ($\dot{\mathbf{q}}_n$) of the platforms are

$$\dot{\mathbf{q}}_a = \begin{bmatrix} \dot{\theta}_{xr_1} \\ \dot{\theta}_s \\ \dot{\theta}_{xr_2} \end{bmatrix}, \quad \dot{\mathbf{q}}_n = \begin{bmatrix} \dot{\theta}_{zr_1} \\ \dot{\theta}_{zr_2} \end{bmatrix} \tag{E.15}$$

and the actuated and non actuated jacobians are

$$\begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_n \end{bmatrix} = \begin{bmatrix} J_{in_a} \\ J_{in_n} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}, \tag{E.16}$$

By substituting in the inverse kinematics genralized solution in equation (A.7) the inverse kinematic solution will be where the actuated inverse kinematic solution is described as

$$\dot{\mathbf{q}}_a = J_{in_a}\ \dot{\mathbf{p}}$$
$$\begin{bmatrix} \dot{\theta}_{xr_1} \\ \dot{\theta}_s \\ \dot{\theta}_{xr_2} \end{bmatrix} = \begin{bmatrix} \frac{-bS(\theta_s)-aC(\theta_s)}{rb} & \frac{bC(\theta_s)-aS(\theta_s)}{rb} & 0 \\ \frac{-C(\theta_s)}{b} & \frac{-S(\theta_s)}{b} & -1 \\ \frac{-bS(\theta_s)+aC(\theta_s)}{rb} & \frac{bC(\theta_s)+aS(\theta_s)}{rb} & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{E.17}$$

While the following sensed forward kinematic solution

$$\dot{\mathbf{p}} = J_{for_s}\ \dot{\mathbf{q}}_s,$$
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{-rbC(\theta_s)-raS(\theta_s)}{2a} & 0 & \frac{rbC(\theta_s)-raS(\theta_s)}{2a} \\ \frac{-rbS(\theta_s)+raC(\theta_s)}{2a} & 0 & \frac{rbS(\theta_s)+raC(\theta_s)}{2a} \\ \frac{r}{2a} & -1 & \frac{-r}{2a} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{xr_1} \\ \dot{\theta}_s \\ \dot{\theta}_{xr_2} \end{bmatrix}. \tag{E.18}$$

E.2 The Robots Dynamics Equations

The simulation process used in chapter (7) requires a dynamic model for each robot. The dynamic model should at least contain the main parameters of the robot for example: the mass (m_p) , inertia (I_p), and the robot mobility constraints. In

this chapter the Euler Lagrange principle is used in deriving the robots dynamic models corresponding to the C3P dynamic model. Since; 1) the wheels parameters have small in comparison the platform parameters, and 2) the platform mobility constraints are more significant than the wheels constrains, the wheels parameter are neglected and the Lagrangian function will be

$$L = K_p, \tag{E.19}$$

where the robot kinetic energy K_p is

$$K_p = \frac{1}{2}\ m_p\ \boldsymbol{V}_p^T\ \boldsymbol{V}_p + \frac{1}{2}\ \Omega_p^T\ I_p\ \Omega_p. \tag{E.20}$$

The robot linear velocities

$$\boldsymbol{V}_p = \begin{bmatrix} V_{px} \\ V_{py} \end{bmatrix}, \tag{E.21}$$

and rotational velocity

$$\Omega_w = \begin{bmatrix} \dot{\phi} \end{bmatrix}, \tag{E.22}$$

are the forward kinematic solutions which carries the mobility constraints that describes the main mobility behavior of the robot. The main function of the dynamic model is to reach a dynamic relation between the WMR actuated torques vector τ and the sensed angular velocities vector $\dot{\mathbf{q}}$.

E.2.1 Holonomic Caster Wheeled Robot

The HCWR actuated and sensed angular velocities of the HCWR are the same, therefore thedynamic model of the robot can be described one equation. Where the linear velocities of the HCWR platform are

$$\boldsymbol{V}_{p(HCWR)} = \frac{1}{3}\left[-\sum_{i=1}^{3}(r\sin(\theta_{s_i})\dot{\theta}_{x_i} + h\ \cos(\alpha_i)\dot{\theta}_{s_i}),\ \sum_{i=1}^{3}(r\cos(\theta_{s_i})\dot{\theta}_{x_i} + h\ \sin(\alpha_i)\dot{\theta}_{s_i})\right]^T \tag{E.23}$$

and the angular velocity is

$$\Omega_{p(HCWR)} = \left[-\sum_{i=1}^{3} \dot{\theta}_{s_i} \right] /3. \tag{E.24}$$

Hence, the Lagrangian function contains the wheel angular velocities $\dot{\mathbf{q}}_x$, steering angular velocities $\dot{\mathbf{q}}_s$, and the steering angles $\mathbf{q}_s$

$$L = f(\mathbf{q}_s, \dot{\mathbf{q}}_s, \dot{\mathbf{q}}_x). \tag{E.25}$$

By substituting the Lagrangian functions in the Lagrangian formulation

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_a}\right) - \frac{\partial L}{\partial \mathbf{q}_a} \tag{E.26}$$

the following dynamic equation is obtained

$$\tau_{(6x1)} = M_H(\mathbf{q})_{(6x6)}\ \ddot{\mathbf{q}}_{(6x1)} + G_{sx_{(6x1)}}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s), \tag{E.27}$$

which can be simplified in the form of

$$\begin{bmatrix} \tau_x \\ \tau_s \end{bmatrix} = \begin{bmatrix} M_{xx_H} & M_{xs_H} \\ M_{sx_H} & M_{ss_H} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_x \\ \ddot{\mathbf{q}}_s \end{bmatrix} + G_{SH_{(6x1)}}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s), \tag{E.28}$$

to separate the torques acting on the wheel velocities from the the torques acting on the angular velocities. The mass matrix $M_H(\mathbf{q})$ is divided in 4 $(3x3)$ matrices

$$M_{xx_H} = \frac{1}{9}\begin{bmatrix} m_p r^2 & m_p r^2 C(\theta_{s_1} - \theta_{s_2}) & m_p r^2 C(\theta_{s_1} - \theta_{s_3}) \\ m_p r^2 C(\theta_{s_2} - \theta_{s_1}) & m_p r^2 & m_p r^2 C(\theta_{s_2} - \theta_{s_3}) \\ m_p r^2 C(\theta_{s_2} - \theta_{s_1}) & m_p r^2 C(\theta_{s_3} - \theta_{s_2}) & m_p r^2 \end{bmatrix}, \tag{E.29}$$

$$M_{xs_H} = \frac{1}{9}\begin{bmatrix} m_p rhC(\alpha_1 - \theta_{s_1}) & m_p rhC(\alpha_2 - \theta_{s_1}) & m_p rhC(\alpha_3 - \theta_{s_1}) \\ m_p rhC(\alpha_1 - \theta_{s_2}) & m_p rhC(\alpha_2 - \theta_{s_2}) & m_p rhC(\alpha_3 - \theta_{s_2}) \\ m_p rhC(\alpha_1 - \theta_{s_3}) & m_p rhC(\alpha_2 - \theta_{s_3}) & m_p rhC(\alpha_3 - \theta_{s_3}) \end{bmatrix}, \tag{E.30}$$

$$M_{sx_H} = \frac{1}{9}\begin{bmatrix} m_p rhC(\alpha_1 - \theta_{s_1}) & m_p rhC(\alpha_1 - \theta_{s_2}) & m_p rhC(\alpha_1 - \theta_{s_3}) \\ m_p rhC(\alpha_2 - \theta_{s_1}) & m_p rhC(\alpha_2 - \theta_{s_2}) & m_p rhC(\alpha_2 - \theta_{s_3}) \\ m_p rhC(\alpha_3 - \theta_{s_1}) & m_p rhC(\alpha_3 - \theta_{s_2}) & m_p rhC(\alpha_3 - \theta_{s_3}) \end{bmatrix}, \tag{E.31}$$

and

$$M_{ss_H} = \frac{1}{9}\begin{bmatrix} m_p h^2 + I_p & m_p h^2 C(\alpha_1 - \alpha_2) + I_p & m_p h^2 C(\alpha_1 - \alpha_3) + I_p \\ m_p h^2 C(\alpha_1 - \alpha_2) + I_p & m_p h^2 + I_p & m_p h^2 C(\alpha_2 - \alpha_3) + I_p \\ m_p h^2 C(\alpha_1 - \alpha_3) + I_p & m_p h^2 C(\alpha_2 - \alpha_3) + I_p & m_p h^2 + I_p \end{bmatrix}. \tag{E.32}$$

The Centripetal and Coriolis velocities $G_{SH}(\dot{\mathbf{q}}_x, \dot{\mathbf{q}}_s, \mathbf{q}_s)$ is a vector of 6 elements, $G_{SH}(1) = 0$, $G_{SH}(2) = 0$, $G_{SH}(3) = 0$,

$$\begin{aligned} G_{SH}(4) = \tfrac{-1}{9}(&m_p rh\ \sin(\alpha_1 - \theta_{x_1})\dot{\theta}_{x_1}\dot{\theta}_{s_1} + m_p rh\ \sin(\alpha_2 - \theta_{x_1})\dot{\theta}_{x_1}\dot{\theta}_{s_2} \\ &+ m_p rh\ \sin(\alpha_3 - \theta_{x_1})\dot{\theta}_{x_1}\dot{\theta}_{s_3} - m_p r^2\ \sin(\theta_{x_1} - \theta_{x_2})\dot{\theta}_{x_1}\dot{\theta}_{x_2} \\ &- m_p r^2\ \sin(\theta_{x_1} - \theta_{x_3})\dot{\theta}_{x_1}\dot{\theta}_{x_3}), \end{aligned} \tag{E.33}$$

$$\begin{aligned} G_{SH}(5) = \tfrac{-1}{9}(&m_p rh\ \sin(\alpha_1 - \theta_{x_2})\dot{\theta}_{x_2}\dot{\theta}_{s_1} + m_p rh\ \sin(\alpha_2 - \theta_{x_2})\dot{\theta}_{x_2}\dot{\theta}_{s_3} \\ &+ m_p rh\ \sin(\alpha_3 - \theta_{x_2})\dot{\theta}_{x_2}\dot{\theta}_{s_3} + m_p r^2\ \sin(\theta_{x_1} - \theta_{x_2})\dot{\theta}_{x_1}\dot{\theta}_{x_2} \\ &- m_p r^2\ \sin(\theta_{x_2} - \theta_{x_3})\dot{\theta}_{x_2}\dot{\theta}_{x_3}), \end{aligned} \tag{E.34}$$

and

$$\begin{aligned} G_{SH}(6) = \tfrac{-1}{9}(&m_p rh\ \sin(\alpha_1 - \theta_{x_3})\dot{\theta}_{x_3}\dot{\theta}_{s_1} + m_p rh\ \sin(\alpha_2 - \theta_{x_3})\dot{\theta}_{x_3}\dot{\theta}_{s_2} \\ &+ m_p rh\ \sin(\alpha_3 - \theta_{x_3})\dot{\theta}_{x_3}\dot{\theta}_{s_3} + m_p r^2\ \sin(\theta_{x_1} - \theta_{x_3})\dot{\theta}_{x_1}\dot{\theta}_{x_3} \\ &+ m_p r^2\ \sin(\theta_{x_2} - \theta_{x_3})\dot{\theta}_{x_2}\dot{\theta}_{x_3}). \end{aligned} \tag{E.35}$$

E.2.2 Omni Directional Wheeled robot

It is noticed from the previous section that the HCWR platform has nonlinear sensed forward solution which delivered nonlinear complex dynamic equation. The ODWR dynamic equation is much more simple because its sensed forward solution is linear as the following

$$\boldsymbol{V}_{p(ODWR)} = R\begin{bmatrix} -\frac{2}{3}\dot{\theta}_{x_1} + \frac{1}{3}\dot{\theta}_{x_2} + \frac{1}{3}\dot{\theta}_{x_3} \\ \frac{1}{\sqrt{3}}\dot{\theta}_{x_2} - \frac{1}{\sqrt{3}}\dot{\theta}_{x_3} \end{bmatrix}, \tag{E.36}$$

and it rotational velocity is

$$\Omega_{p(ODWR)} = \frac{R}{3h}\left[-\sum_{i=1}^{3}\dot{\theta}_{x_i}\right]. \tag{E.37}$$

Since the Lagrangian function is a function of the actuated wheels velocities $\dot{\mathbf{q}}_x$ as shown

$$L = f(\dot{\mathbf{q}}_x), \tag{E.38}$$

the dynamic equation of the robot platform is

$$\tau_{o_{(3x1)}} = M_{o_{(3x3)}}\ \ddot{\mathbf{q}}_{a_{(3x1)}}, \tag{E.39}$$

which can be detailed in the following equation

$$M_{o_{(3x3)}} = \frac{R^2}{9h^2}\begin{bmatrix} 4m_p\ h^2 + I_p & -2m_p\ h^2 + I_p & -2m_p\ h^2 + I_p \\ -2m_p\ h^2 + I_p & m_p\ h^2 + 3m_p\frac{h^2}{R^2} + I_p & m_p\ h^2 - 3m_p\frac{h^2}{R^2} + I_p \\ -2m_p\ h^2 + I_p & m_p\ h^2 - 3m_p\frac{h^2}{R^2} + I_p & m_p\ h^2 + 3m_p\frac{h^2}{R^2} + I_p \end{bmatrix} \tag{E.40}$$

E.2.3 Ramsis II

The mobile robot Ramsis II had been not model dynamically before. In this section a dynamic model is proposed using the forward kinematic solution for the robot velocities

$$\boldsymbol{V}_{p(RII)} = \begin{bmatrix} \frac{-rbC(\theta_s)-raS(\theta_s)}{2a}\dot{\theta}_{xr_1} + \frac{rbC(\theta_s)-raS(\theta_s)}{2a}\dot{\theta}_{xr_2} \\ \frac{-rbC(\theta_s)-raS(\theta_s)}{2a}\dot{\theta}_{xr_1} + \frac{rbC(\theta_s)-raS(\theta_s)}{2a}\dot{\theta}_{xr_2} \end{bmatrix}, \tag{E.41}$$

and

$$\Omega_{p(RII)} = \begin{bmatrix} \frac{r}{2a}\dot{\theta}_{x_1} - \dot{\theta}_s + \frac{-r}{2a}\dot{\theta}_{x_2} \end{bmatrix}. \tag{E.42}$$

The robot velocities are substituted in the kinetic energy equation (E.20) to obtain the Lagrangian function which is function of

$$L = f(\dot{\theta}_{x_1}, \dot{\theta}_{x_s}, \dot{\theta}_s). \tag{E.43}$$

which is used to derive the dynamic equation with the Lagrangian formulation

of equation (E.27). As a result the following dynamic relation is obtained

$$\tau_{(3x1)} = M_{(3x3)}\ \ddot{\mathbf{q}}_{a_{(3x1)}}, \tag{E.44}$$

where the Centripetal and Coriolis velocities are zeros because the mass and inertia of the actuators were neglected due to their very small values compared to platform mass and inertia. The dynamic relation (E.44) can be written in more details as the following

$$\begin{bmatrix} \tau_{xr_1} \\ \tau_s \\ \tau_{xr_2} \end{bmatrix} = \begin{bmatrix} \frac{m_p r^2(a^2+b^2)+I_p r^2}{4a^2} & -\frac{rI_p}{2a} & \frac{m_p r^2(a^2-b^2)-I_p r^2}{4a^2} \\ -\frac{rI_p}{2a} & I_p & \frac{rI_p}{2a} \\ \frac{m_p r^2(a^2-b^2)-I_p r^2}{4a^2} & \frac{rI_p}{2a} & \frac{m_p r^2(a^2+b^2)+I_p r^2}{4a^2} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_{x_1} \\ \ddot{\theta}_s \\ \ddot{\theta}_{x_2} \end{bmatrix} \tag{E.45}$$

Bibliography

[1] H. Gross and H. Boehme, " Perese-a vision-based interactive mobile shopping assistant," *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, Vol.1, 2000, pp: 80-85 vol.

[2] T. Kiriki, Y. Kimuro and T. Hasegawa, " A 4-legged mobile robot control to observe a human behavior," *Proceedings of the 1999 IEEE International Workshop on Robot and Human Interaction*, Pisa, Italy, September 1999.

[3] G. Dudek and M. Jenkin, " Computational Principles of Mobile Robotics," Cambridge University Press,2000.

[4] Muir, P. " Modeling and control of wheeled mobile robots," *PhD dissertation Carnegie mellon university* (1987).

[5] W. Burgard, P. Trahanias, D. Haehnel, M. Moors, D. Schulz, H. Baltzakis, A. Argyros ,"TOURBOT and WebFAIR: Web-Operated Mobile Robots for Tele-Presence in Populated Exhibitions," *IEEE/RSJ Conf. IROS'02, Full day workshop in "Robots in Exhibitions"*, Lausanne, Switzerland, October 1, 2002.

[6] G. Steinbauer and F. Wotawa,"Mobile Robots in Exhibitions, Games and Education. Do They Really Help?," *CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby*, Vienna, Austria, December 2 4, 2004.

[7] M. Bennewitz, F. Faber, D. Joho, M. Schreiber, and S. Behnke, "Towards a Humanoid Museum Guide Robot that Interacts with Multiple Persons," *Pro-*

ceedings of 5th IEEE-RAS International Conference on Humanoid Robots, 2005
.

[8] U. Kartoun, H. Stern, Y. Edan, C. Feied, J. Handler, M. Smith and M. Gillam, "Vision-Based Autonomous Robot Self-Docking and Recharging," *11th International Symposium on Robotics and Applications (ISORA)*, Budapest, Hungary, July 24 - 27, 2006.

[9] J. Bruemmer, J. Marble and D. Dudenhoffer,"Intelligent Robots for Use in Hazardous DOR Environments," *Idaho National Engineering and Environmental Laboratory*, 1998.

[10] P. Chakravarty, D. Rawlinson and R. Jarvis, "Distributed Visual Servoing of a Mobile Robot for Surveillance Applications," *Australasian Conference on Robotics and Automation (ACRA)*, 2004.

[11] P. Biber, S. Fleck and T. Duckett, "3D Modeling of Indoor Environments by a Mobile Platform with a Laser Scanner and Panoramic Camera," Proc. 13th European Signal Processing Conference (EUSIPCO 2005).

[12] X. Yun, and N. Sarkar, "Unified formulation of robotic systems with holonomic and nonholonomic constrains," *IEEE Trans. on robotics and automation*, Vol.12, 1998, pp.640-650.

[13] Y. Peng, and E. Badreddin. "Analyses and Simulation of the Kinematics of A Wheeled Mobile Robot Platform with Three Castor Wheels." *Internal Report Automation Lab, Computer Engineering University of Mannheim*, 2000.

[14] R. Featherstone, D. Orin, "Robot Dynamics: Equations and Algorithms," *IEEE International Conference on Robotics and Automation*, 2000, pp.826-834.

[15] M. I. C. Dede, and S. Tosunoglu, "Design of a Fault-Tolerant Holonomic Mobile Platform," *Florida Conference on Recent Advances in Robotics, FCRAR 2006*,Miami, Florida, 2006.

[16] M. Jung, J. Jang and J. Kim, "Development of RoboSot Category Robot Soccer Team," *Dept. of Electrical Engineering and Computer Science, KAIST, Taejon-shi*, pp.305-701.

[17] Souma M. Alhaj Ali and Ernest L. Hall, "Designing and simulation a motion Controller for a Wheeled Mobile Robot Autonomous Navigation." *Proc. of SPIE Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision*,Vol. 6006. 2005.

[18] L. Huang, Y. S. Lim, David Li and E. L. Teoh, " Desgin and Analysis of a four-wheel Omnidirectional Mobile Robot." *2nd international conference on Autonomous Robots and Agents*,New Zealand. 2004.

[19] Essam Badreddin, " A Hybrid Control Structure for a Robot Soccer Player." *Proceedings of world Automation Congress*, Maui, 2000.

[20] Joaquin Sitte and Petra Winzer, " Mastering complexity in robot design." *Proceedings of international conference on intelligent robots and systems*, Japan, 2004.

[21] Peter Xu, " Mechatronics Design of a Mecanum Wheeled Mobile Robot." *Cutting Edge Robotics*, Germany, pp:784 2005.

[22] C. R. Fulmer, "Design and fabrication of an Omnidirectional Vehicle platform", *Master thesis University of Florida*, Florida, 2003.

[23] K.L. Moore, N.S. Flann ," Six-Wheeled Omnidirectional Autonomous Mobile Robot," *IEEE Control Systems Magazine*, December, 2000, pp. 53-66.

[24] A. Yamashita, H. Asama, H. Kaetsu, I. Endo and T. Arai, "Development of Omni-Directional and Step-Climbing Mobile Robot", *Proceedings of the 3rd International Conference on Field and Service Robotics (FSR2001)* Espoo (Finland), June 2001, pp.327-332.

[25] D. Chugo, K. Kmubata, H. Kuetsu, H. Asama and T. Mishimu, "Development of Omni-Directional Vehicle with Step-Climbing Ability ", *Proceedings of the 2003 IEEE International Conference on Robotics & Automation* Taipei, Taiwan, September 14-19,2003.

[26] H. Yu, S. Dubowsky and A. Skwersky, "Omni-Directional Mobility Using Active Split Offset Castors", *Journal of Mechanical Design* Vol. 126, September 2004, pp:822-829.

[27] J. Michel, E. Colgate and W. Wannasuphorasit, "Cobot architecture," Northwetern university, 1999.

[28] J. E. M. Salih, M. Rizon, S. Yaacob, A. Adom and M. R. Mamat ,"Designing Omni-Directional Mobile Robot with Mecanum Wheel," *American Journal of Applied Sciences*, Vol 3(5), 2006, pp.1831-1835.

[29] Robert C. McCord, "Snap-Fit Plastic Caster," United states Patent 6588059 B1, July 2003.

[30] John D.Bearden, "LAWN Mower Adjustable Wheel Conversion Assembly," United States Patent 6899345 B1, May 2005.

[31] D.Wang and G Xu , "Full-State Tracking and Internal Dynamics of Nonholonomic Wheeled Mobile Robots," *IEEE Transactions on Mechatronics*, Vol. 8, No.2, June 2003.

[32] Li, Y.P, T.Zielinska, M.H.Jr Ang and W.Lin, " Vehicle Dynamics of Redundant Mobile Robots with Powered Caster Wheels", *Proceedings of the Sixteenth CISM-IFToMM Symposium, Romansy 16, Robot Design, Dynamics and Control*, ed.Teresa Zielinska and Cezary Zielinski (2006): 221-228. Warsaw, Springer.

[33] M. Wada and S. Mori, " Holonomic and omnidirectional vehicle with conventional tires," *Proc. IEEE Intl. Conf. Robotics and Automation*,4:3671-3676, 1996.

[34] B.J. Yi and W.K. Kim, " The kinematics for redundantly actuated omnidirectional mobile robots," *J.Robotic Systems*, 12(6):255-267, 2002.

[35] Y.P. Li, D.N. Oetomo, Marcelo H. Ang Jr., and C.W. Lim, " Torque distribution and slip minimization in an omnidirectional mobile base," *Intl. Conf. Advanced Robotics*, pages 567-572, 2005.

[36] T.Z. Maung, D. Oetomo, M.H. Ang Jr., T.K. Ng, " Kinematics and Dynamics of an Omnidirectional Mobile Platform with Powered Caster Wheels," *International Symposium on Dynamics and Control, Hanoi, Vietnam*, 15-20 September, 2003

[37] D. Kim, H. Lee and W. Kwon, " Geometeric Kinematics Modeling of omnidirectional autonomous Mobile robot and its application," *International conference on robotics and automation*, San Francisco CA, 2000.

[38] R. Holmberg and O. Khatib, " Development of Holonomic Mobile Robot for Mobile Manipulation Tasks," *FSR99 Int. conference on field and services robotics*, Pittsburgh, 1999.

[39] J. Chung, B. Yi , W. Kim and H. Lee ," The Dynamic Modeling and Analysis for An Omnidirectional Mobile Robot with Three Caster Wheels," *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, 2003.

[40] W. O. Schiehlen, " Multibody System Handbook," Springer-Verlag, 1990.

[41] Khatib, H. Jaouni, R. Chatila and J. P. Laumond, " Dynamic Path Modification for Car-Like Nonholonomic Mobile Robots," *In: IEEE Intl. Conf. on Robotics and Automation*, 1997, pp. 29202925.

[42] G. Ramrez and S. Zeghloul, " A New Local Path Planner for Nonholonmic Mobile Robot Navigation in Cluttered Environments," *In: IEEE Int. Conf. on Robotics and Automation*, 2000.

[43] G. Campion, G. Bastin, and B. DAndrea-Novel, " Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans Roboti Automa*12:(1) 1996, pp.4762.

[44] Wheekuk Kim, Yongi-Kun, and Dong Jin Lim, " Kinematic Modeling of Mobile Robots by Transfer Method of Augmented Generalized Coordinates," *Journal of Robotic Systems* 21(6), 2004, pp.301322.

[45] Javier Garcia de Jalon, " Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challenge", ISBN 0-387-94096-0, 440 pp., Springer-Verlag, New-York, 1994.

[46] A. Albagual and Wahyudi, " Dynamic Modeling and Adaptive Traction Control for Mobile Robots," *International journal or advanced robotic systems,* Volum 1, number 3, 2004.

[47] G. B. Chung, B. Yi, D.J.Lim, and W. Kim, " An Efficient Dynamic Modeling Methodology for General Type of Hybrid Robotic Systems," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, New Orleans, LA April 2004.

[48] Karim A. Tahboub and Harry H. Asada, " Dynamics Analysis and Control of a Holonomic Vehicle With a Continuously Variable Transmission," *Transactions of the ASME*, Vol. 124, March 2002, pp.118-126.

[49] Robert L. Williams II, Brian E. Carter, Paolo Gallina, and Giulio Rosati, " Dynamic Model with Slip for Wheeled Omni-Directional Robots," *Final Manuscript, IEEE Transaction on Robotics and Automation*, March 2002.

[50] , Z. Huang, N. M. Thalmann, D. Thalmann, " Interactive human motion control using a closed-form of direct and inverse dynamics," *Proceeding of the second Pacific conference on Fundamentals of computer graphics* ,Beijing, China , 1995, pp.243-255.

[51] M. Li, T. Huang, J. Mei, and X. Zhao, " Dynamic Formulation and Performance Comparison of the 3-DOF Modules of Two Reconfigurable PKMthe Tricept and the TriVariant," *Journal of Mechanical Design*, Vol. 127, November 2005, pp.1129-1136.

[52] A. C. Fang and N. S. Pollard, " Efficient Synthesis of Physically Valid Human Motion," *ACM SIGGRAPH 2003 conference proceedings*, San Diego, 2003.

[53] E. Anli, H. Alp, S. N. Yurt and I. Ozkol, " The Stewart Platform Mechanism A Review," *Transactions on Engineering, Computer and Technology* Vol. 2 December 2004, pp:106-109.

[54] E. Papadopoulos, R. Frenette, B. Mu, and Y. Gonthier, " On the Modeling and Control of an Experimental Harvester Machine Manipulator," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* Grenoble, France, September 8-12 1997.

[55] N. Chakraborty and A. Ghosal, " Dynamic modeling and simulation of a wheeled mobile robot for traversing uneven terrain without slip," *Journal of Mechanical Design*, Volume 127, Issue 5, September 2005, pp. 901-909 .

[56] David Baraff, " Linear-time dynamics using Lagrange multipliers," *International Conference on Computer Graphics and Interactive Techniques* , 1996, pp.137-146.

[57] J. Naudet and D. Lefeber, " Recursive algorithm based on canonical momenta for forward dynamics of multibody Systems," *Proceedings of IDETC/CIE 2005 ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* September 24-28, 2005, Long Beach, USA.

[58] M. zefran and F. Bullo, " Lagrangian Dynamics," *Robotics and Automation Handbook* Chapter 1, T. R. Kurfess, 2005.

[59] S. Bruder and K. Wedeward, " An interactive online robotics course ," *Intelligent Automation and Soft Computing* Vol. 13, No. X,, 2007, pp. 1-12.

[60] I. F. Ihle, J. Jouffroy and T. I. Fossen, " Formation Control of Marine Surface Craft using Lagrange Multipliers ," *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005* Seville, Spain, December 12-15, 2005.

[61] J.H. Suh, J.W. Lee, and K.S. Lee, " Modeling and Control of Cooperative 3-Wheeled Mobile Robots with Decentralized Passive Velocity Field Control," *Proceeding (412) Modelling, Identification, and Control*, 2004.

[62] E. Badreddin, " Fuzzy relations for behavior-fusion of mobile robots," *Proceedings of the IEEE Conference on Robotics and Automation* San Diego, California, May 8-13, 1994.

[63] Essameddin Badreddin , "Autonomous mobile robots," Lecture documentation Mannheim university, Chapter 4.

[64] Essameddin Badreddin , "Recursive Control Structure for Mobile Robots," *International Conf. on Intelligent Autonomous Systems.2* Amesterdam, 11-14 Dec, 1987.

[65] Essameddin Badreddin , "Recursive Behavior-Based Architecture for Mobile Robots," *Robotics and Autonomous Systems* Vol. 8, 1991.

[66] R. W. Brokett , "Asymptotic Stability and Feedback Stabilization," *Differential Geometric cControl Theory*,Berlin, Germany : Brikhauser, 1983, pp:181-191.

[67] J. Zabczyk, "Some Comments on Stabilizability," *Int. J. App. Math. Aptim.*, Vol. 19, 1989, pp:1-9.

[68] P. Biber, S. Fleck and T. Duckett, "3D Modeling of Indoor Environments by a Mobile Platform with a Laser Scanner and Panoramic Camera," *Proc. 13th European Signal Processing Conference (EUSIPCO 2005)*,2005.

[69] A. El-Shenawy,A. Wagner, and E. Badreddin . "Solving The Singularity Problem for a Holonomic Mobile," *4th IFAC-Symposium on Mechatronic Systems (MECHATRONICS 2006)* , Germany, 2006.

[70] A. El-Shenawy,A. Wagner, and E. Badreddin . "Controlling a Holonomic Mobile Robot With Kinematics Singularities." *The 6th World Congress on Intelligent Control and Automation*, China, 2006.

[71] A. El-Shenawy,A. Wagner, and E. Badreddin . "Dynamic Model of a Holonomic Mobile Robot with Actuated Caster Wheels." *The 9th International Conference on Control, Automation, Robotics and Vision, ICARCV* , Singapore, 2006.

[72] A. El-Shenawy,A. Wagner, and E. Badreddin . "Inverse Dynamic Solution for Holonomic Wheeled Mobile Robot with Modular Actuation." *The European Control Conference, ECC07*, Kos, Greece, 2007.

[73] A. El-Shenawy,A. Wagner, and E. Badreddin . "Kinematics and Dynamics Analysis for a Holonomic Wheeled Mobile Robot." *ICINCO 2007, Angers, France*, May , 2007.

[74] A. El-Shenawy,A. Wagner, and E. Badreddin . "Practical Construction and Position Control of a Modular Actuated Holonomic Wheeled Mobile Robot." *International Conference for Robotics and Automation ICRA08, California, USA*, May , 2008.

[75] A. El-Shenawy, A. Wellenreuther, A. Baumgart, and E. Badreddin ."Comparing Different Holonomic Mobile Robots ."*IEEE International Conference on Systems, Man, and Cybernetics, Montreal* , October , 2007.

[76] A. Albagul and Wahyudi, "Dynamic Modelling and Adaptive Traction control for Mobile Robots," *Int. Jor. of Advanced Robotic Systems,* Vol.1,no.3 (2004), pp.1729-8806.

[77] J. R. Asensio and L. Montano, "A Kinematic and Dynamic Model-Based Motion Controller for Mobile Robots,"*15th Triennial World Congress,* Barcelona, Spain, 2002.

[78] J. Naudet and D. Lefeber, "Recursive Algorithm Based on Canonical Momenta For Forward dynamics of Multibody Systems," *Proceedings of IDETC/CIE 2005*, 2005.

[79] P. Muir and C. P. Neuman, "Kinematic modeling of wheeled mobile robots," *Journal of Robotic system* , Vol4 Issue 2, pp-281-340. Mar 2007

[80] L. Garcia and J. Tornero, "Kinematic control of wheeled mobile robots," *Latin American Applied Research*, Vol 38 pp.7-16, 2008.

[81] JC Habumuremyi, Jonathan Houpin, "Model of a wheeled robot named Robudem and design of a state feedback controller for its posture tracking: simulation and experiment," *ISMCR05*, 8-10 November 2005, Brussels, Belgium.

[82] D M. helmick, S. I. Roumeliotis, Y. cheng, "Slip-compensated path following for planetary exploration rovers,"*Advanced Robotics* , Vol. 20, No. 11, pp. 1257-1280 (2006).

[83] R.M.Kiehn ,"Holonomic and anholonomic constraints and coordinates, Frobenius integrability and torsion of various," Emeritus, Phys Dept., University of huston, 2001.

[84] N. Sarkar, X. Yun, and V. Kumar, "Control of mechanical systems with rolling contacts: Applications to dynamic control of mobile robots, Int. J. Robot. Res., vol. 13, no. 1, 1994. *Int. J. Robot. Res.*, vol. 13, no. 1, 1994.

[85] A.lessandro De Luca, G.iuseppe Oriolo, P.aolo R.obuffo Giordan,"Kinematic Modeling and Redundancy Resolution for Nonholonomic Mobile Manipulators," *Proceedings of the 2006 IEEE International Conference on Robotics and Automation Orlando, Florida* , May 2006

[86] P.A.J. Koenders, "On the relation between equation formulation of constrained systems and implicit numerical integration and optimisation," *M.Sc. Thesis, Faculty of Electrical Engineering, University of Twente,* Netherland June 2002.

[87] J. Kim, J. K. Mills, and M. R. Popovic, "The Inverse Dynamics Solutions of Arm-Free Standing for Paraplegics," *10th Annual Conference of the International FES Society*, Montreal, Canada, July 2005 .

[88] R. Featherstone and D. Orin, "Robot Dynamics: Equations and Algorithms," *Proc. IEEE Int. Conf. Robotics & Automation,* San Francisco, CA, 2000, pp. 826-834 .

[89] G. Rodriguez, "Kalman Filtering, Smoothing, and Recursive Robot Arm Forward and Inverse Dynamics," *IEEE Journal on Robotics and Automation,* vol. RA-3, no. 6, 1987, pp. 624-639.

[90] W.E. Dixon, W.E. Galluzo, G. Hu and C. Crane, "Adaptive velocity field control of a wheeled mobile robot," *Proceedings of the Fifth International Workshop on Robot Motion and Control,* 23-25 June 2005.

[91] T. Jin, J. Lee, and H. Hashimoto, "Position Control of Mobile Robot for Human-Following in Intelligent Space with Distributed Sensors," *International Journal of Control, Automation, and Systems,* vol. 4, No. 2, April 2006, pp. 204-216.

[92] A.M.Lyapunov, "The general problem of the stability of motion," *Int. J. Control*, 1992, Vol.55, No.3, pp 531-773.

[93] Tai-Yu Wang,Ching-Chih Tsai, "Adaptive Trajectory Tracking Control of a Wheeled Mobile Robot via Lyapunov Techniques," *The 30th Annual Conference of the IEEE lnduslrfal Electronics Society*, Busan, Korea November 2 - 6,2004.

[94] Krzysztof Kozlowsk, Dariusz Pazderski, "Modeling and Control of a 4-Wheele Skid-Steering Mobile Robot," *TInt. J. Appl. Math. Comput. Sci., 2004*,Vol. 14. No. 2, pp. 143160, 2007.

[95] Bibhya Sharma and Jito Vanualailai, "Lyapunov Stability of a Nonholonomic Car-like Robotic System," *Appeared in Nonlinear Studies*,Vol. 14, No. 4, 477496, 2004.

[96] M.D. Adams, "Adaptive Motor Control to Aid Mobile Robot Trajectory Execution in the Presence of Changing System Parameters," *IEEE Transactions on Robotics and Automation,* Vol 14, No.6, December 1998, pp.894-901.

[97] C. M. Soria, R. Carelli, R. Kelly, and J. M. I. Zannatha, "Coordinated Control Of Mobile Robots Based On Artificial Vision," *International Journal of Computers, Communications & Control,* Vol. I, No. 2, 2006, pp. 85-94.

[98] B. Carter, M. Good, M. Corohoff, J. Lew , "Mechanical Design and Modeling of an Omni-directional RoboCup Player." *Proceeding RoboCup 2001*. International Symposium Seattle, August, 2001.

[99] R. Hafner and M. Riedmiller, "Reinforcement Learning on a Omnidirectional Mobile Robot." *IEEE/RSJ International Conference on Intelligent Robots and Systems for Human Security, Health, and Prosperity*, 2003.

[100] K. Watanabe, Y. Shiraishi, S. Tzafestas, J. Tang, and T. Fukuda, "Feedback Control of an Omnidirectional Autonomous Platform for Mobile Service Robots." *Journal of Intelligent and Robotic Systems*. Vol 22: 315-330, 1998.

[101] M.J. Jung, H. S. Kim, S. Kim, and J. H. Kim, 2000, "Omni-Directional Mobile Base OK-II", *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol.4 pp.3449-3454.

[102] Essameddin Badreddin , "Autonomous mobile robots," Lecture documentation Mannheim university, Chapter 9.

[103] Mark Steven Tisius , "An Empirical Approach to Performance Criteria and Redundancy Resolution," Doctor of Philosophy Dissertation Presented to the Faculty of the Graduate School of The University of Texas at Austin, 2004.

[104] Christopher David Cocca , "Failure Recovery in Redundant Serial Manipulators," Doctor of Philosophy Dissertation Presented to the Faculty of the Graduate School of The University of Texas at Austin, 2000.

[105] Mitchell Wayne Pryor , "Complex Task Completion with Redundant Serial Manipulators," Masters Thesis Presented to the Faculty of the Graduate School of The University of Texas at Austin, 1999.

[106] Chalongrath Pholsiri , "Task-based Decision Mmaking and Control of Robotic Manipulators," Doctor of Philosophy Dissertation Presented to the Faculty of the Graduate School of the University of Texas at Austin, 2004.

[107] K.M.Miettinen. "Nonlinear Multiobjective Optimization." Kluwer Academic Publishers, London, 1999.

[108] D. Kim, H. Lee, and W. Kwon, "Geometric Kinematics Modeling of Omnidirectional Autonomous Mobile Robot and Its Applications." *Proceeding of the 2000 IEE international Conference on Robotics and Automation.* San Francisco, CA. April 2000.

[109] W. Kim, B.J Yi, and D. J. Lim, "Kinematic Modeling of Mobile robots by Transfer Method of Augmented Generalized Co-ordinates." *Journal of Robotic Systems.* Wiley InterScince 21(6), 2004, pp:301-322.

[110] R. Rojas and A. G. Foerster, "Holonomic Control of a Robot with an Omnidirectional Drive." *Kuenstliche Intelligenz.* BoettcherIT Verlag, 2006.

[111] D. Chugo, K. Kawabata, H. Kaetsu, H. Asama and Taketoshinismia, "Development of a control system for an omni-directional vehicle with step-climbing ability", *Advanced Robotics,* Vol. 19, No. 1, 2005, pp. 5571.

[112] H. Samani, A. Abdollahi, Hossein Ostadi, and S.Z. Rad, "Design and Development of a Comprehensive Omni directional Soccer Player Robot",*International Journal of Advanced Robotic Systems ,* Vol. 1, No. 3, Sept 2004, pp. 191-200.

[113] S. Rad, F. Sharifi, M. D. Panah, A. Abdollahi, H. Ostadi and H. Samani, "A Practical Approach to Control and Self-Localization of Persia Omni-Directional

Mobile Robot",*IROS/IEEE International Conference on Intelligent Robotic Systems,* Edmonton, Alberta, Canada 2005.

[114] O.Diegel, A. Badve, G. Bright, J. Potgeiter, and S. Tlale, "Improved Mecanum Wheel Design for Omni-directional Robots." *Proc. Australian Conference on Robotics and Automation.* Auckland, pp:27-29, November 2002.

[115] J. Boreinstein, H.R. Everett, and L. Feng, "Navigating Mobile Robot." *Ak, Peters, Wellesley*, Massachusetts,1996.

[116] W. Nadir, I. Y. Kim, and O. L. Weck. "Structural Shape Optimization Considering Both Performance and Manufacturing Cost." *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, USA, 2004.

[117] O. L. Weck and I. Y. Kim. "Adaptive Weighted Sum Method for Bi-objective Optimization." *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, California, USA, 2004.

[118] Y.Lui, X. Wu, J.J. Zhu and J. Lew . "Omni-Directional Mobile Controller Design by Trajectory Linearization." *In Procceding 2003 American Control Conference, ACC 2003*, Denver, Colorado,2003.